

**Titre:** Détermination d'une métrique pour le problème de localisation et  
cartographie simultanées  
Title:

**Auteur:** Alexandre Aubert  
Author:

**Date:** 2013

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Aubert, A. (2013). Détermination d'une métrique pour le problème de localisation  
et cartographie simultanées [Mémoire de maîtrise, École Polytechnique de  
Montréal]. PolyPublie. <https://publications.polymtl.ca/1227/>  
Citation:

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/1227/>  
PolyPublie URL:

**Directeurs de  
recherche:** Richard Gourdeau  
Advisors:

**Programme:** génie électrique  
Program:

UNIVERSITÉ DE MONTRÉAL

DÉTERMINATION D'UNE MÉTRIQUE POUR LE PROBLÈME DE LOCALISATION  
ET CARTOGRAPHIE SIMULTANÉES

ALEXANDRE AUBERT  
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE ÉLECTRIQUE)  
AOÛT 2013

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉTERMINATION D'UNE MÉTRIQUE POUR LE PROBLÈME DE LOCALISATION  
ET CARTOGRAPHIE SIMULTANÉES

présenté par : AUBERT Alexandre

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

a été dûment accepté par le jury d'examen constitué de :

M. SAUSSIÉ David, Ph.D., président

M. GOURDEAU Richard, Ph.D., membre et directeur de recherche

M. GEMME Sébastien, M.Sc.A., membre

## REMERCIEMENTS

Je tiens à remercier l'Agence Spatiale Canadienne qui m'a permis d'utiliser leurs installations ainsi que leur base de données au cours de ce projet. Un merci spécial à Monsieur Sébastien Gemme qui a su me procurer des conseils importants pour mes travaux.

Je suis aussi très reconnaissant envers le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) qui m'a octroyé une bourse d'étude me permettant de me concentrer sur ce projet.

Mes remerciements s'adressent aussi à Monsieur Richard Gourdeau, mon directeur de recherche, qui m'a guidé et soutenu tout au long de ma maîtrise. De plus, il m'a offert plusieurs opportunités pour m'impliquer dans des charges de laboratoires à l'École Polytechnique de Montréal. J'ai donc eu la chance d'acquérir un peu d'expérience en enseignement, un domaine qui va de pair avec la recherche.



## RÉSUMÉ

Les robots mobiles ont une autonomie limitée lorsqu'ils se retrouvent en environnement inconnu. Ils doivent alors construire une carte de leur environnement tout en s'y situant en même temps. Cette carte servira par la suite à la planification de trajectoire pour que le robot puisse effectuer sa tâche. Cette carte globale se doit donc d'être la plus précise possible, sinon le robot pourrait rencontrer des obstacles nuisibles et ne jamais atteindre la destination désirée. Le processus pour obtenir cette carte est composé d'une première phase d'acquisition où le robot utilise un LIDAR lui permettant de modéliser son environnement immédiat par un nuage de points. Ensuite, il faut relier les acquisitions entre elles le plus fidèlement possible pour obtenir un nuage de points couvrant une plus grande partie de l'environnement.

Les balayages sont d'abord regroupés deux par deux par un algorithme de recalage appelé ICP. Cette technique est très efficace lorsque l'estimé initial de la position est près du résultat recherché. Par contre, lorsque ces conditions initiales sont plus erronées, il est fortement possible de converger vers un minimum local. Cette erreur de recalage se propage ensuite lorsque le robot continue sa trajectoire. Il est donc important de trouver une technique permettant de qualifier le résultat d'un recalage et ainsi de rejeter les mauvaises informations.

La métrique proposée dans ce travail est un algorithme de discrimination où plusieurs paramètres sont comparés entre eux. Une base de données composée de plusieurs bons et mauvais résultats séparés en deux groupes respectifs a été créée. Elle permet ensuite de comparer un nouveau recalage par rapport à ces deux groupes et d'attribuer une probabilité d'appartenance à chacun des groupes. Ces valeurs donnent alors une indication de la fiabilité du recalage. Les résultats obtenus donnent un taux d'erreur de classification de 2.12%.

Cette métrique est par la suite utilisée pour corriger l'accumulation d'erreurs au cours d'une longue trajectoire. En effet, lorsque le robot revisite un endroit déjà modélisé, il obtient de nouvelles informations permettant d'effectuer une correction sur sa position estimée et ainsi de réduire l'erreur accumulée. Ceci est fait à l'aide d'un algorithme de distribution d'erreur où le lien entre chaque paire de balayages est modélisé par des ressorts. L'ensemble des ressorts et des balayages constitue alors un système dynamique qui aura tendance à converger vers une position stable qui correspond au minimum d'énergie. Les positions estimées des balayages seront alors modifiées et l'erreur accumulée sera corrigée pour obtenir une carte globale de l'environnement qui est plus uniforme. La métrique de recalage permet de déterminer les constantes de raideurs des ressorts. Ainsi, les recalages moins fiables seront représentés par des ressorts plus flexibles qui pourront s'étirer pour mieux respecter les contraintes des ressorts plus rigides correspondant aux bons recalages.

## ABSTRACT

Mobile robots have a limited autonomy when they are placed in an unknown environment. They must simultaneously create a map and position themselves on that map. Then the robot will be able to plan his trajectory to avoid obstacles and accomplish its task. The global map must be precise, otherwise the robot could hit an obstacle or get lost and not reach his destination. The first step to obtain this global map is to use a LIDAR to scan the immediate environment and modelize it by a scatter plot. Then, scans are registered together to create a bigger scatter plot and so on.

The scans are first registered two by two with a registration algorithm called ICP. This algorithm leads to a very good estimate of the position if the initial conditions are good enough. Otherwise, it may converge towards a local minima and introduce an error in the global map. This error is then propagated along the robot path. Thus, it is very important to qualify the registration result to be able to reject false informations.

This work introduces a discrimination algorithm which compares specific parameters to obtain a registration metric. A database composed of multiple registration results has been created. It is composed of two distinct groups which are the good and the bad results. Then, a new registration result can be compared against these groups to see which ones it relates the most to. The probability of appartenance of this new data to each group gives the probability that the registration converged toward the solution. The parameters used in the discrimination algorithm give an apparent error rate of 2.12%.

This metric is then used to minimize the error accumulation during long trajectories. When the robot reaches a previously visited location, it can acquire new data and this new information can be used to correct its estimated position and reduce the errors accumulated during its journey. An error distribution error is used to make this correction. Virtual springs are connected between each pair of scans. Then, the springs and scans constitute a dynamic system which naturally tends towards a stable position consisting of an energy minimum. The new scans position consist of a more consistent map where the error is equally distributed. The registration metric can be used to assign the stiffness of the springs. A less reliable registration result leads to a less stiff spring which can stretch to meet the constraints of the other springs representing better registration results.

## TABLE DES MATIÈRES

REMERCIEMENTS . . . . .	iii
RÉSUMÉ . . . . .	iv
ABSTRACT . . . . .	v
TABLE DES MATIÈRES . . . . .	vi
LISTE DES TABLEAUX . . . . .	viii
LISTE DES FIGURES . . . . .	ix
LISTE DES SIGLES ET ABRÉVIATIONS . . . . .	xi
CHAPITRE 1 INTRODUCTION . . . . .	1
1.1 Éléments de la problématique . . . . .	1
1.2 Objectifs de recherche . . . . .	4
1.3 Plan du mémoire . . . . .	4
CHAPITRE 2 REVUE DE LITTÉRATURE . . . . .	7
2.1 Algorithme ICP et variantes . . . . .	7
2.2 Recalage de plusieurs surfaces . . . . .	8
2.3 Applications spécifiques en robotique mobile . . . . .	8
CHAPITRE 3 ALGORITHME DE BOUCLE FERMÉE . . . . .	10
3.1 Nombre et emplacement des ressorts . . . . .	12
3.2 Algorithme de répartition de l'erreur . . . . .	12
3.3 Force des ressorts . . . . .	13
CHAPITRE 4 DÉTERMINATION D'UNE MÉTRIQUE . . . . .	18
4.1 Création de la base de données . . . . .	18
4.2 Algorithme de discrimination . . . . .	19
4.3 Variables . . . . .	19
4.3.1 Pourcentage de superposition . . . . .	20
4.3.2 Nombre d'itérations . . . . .	21
4.3.3 Relief des balayages . . . . .	22

4.3.4	Distance de Hausdorff . . . . .	25
4.3.5	« Spin image » . . . . .	26
4.3.6	Image de distances (« Range image ») . . . . .	29
4.3.7	Distance de chaque point avec son plus proche voisin . . . . .	31
CHAPITRE 5 DONNÉES EXPÉRIMENTALES . . . . .		33
5.1	Terrain d'émulation de Mars . . . . .	33
5.1.1	« Saddle Mountain » . . . . .	33
5.1.2	Plaque de dalles Alpha . . . . .	34
5.2	Données de « Mobile Robotics Test-Bed » . . . . .	35
CHAPITRE 6 RÉSULTATS . . . . .		37
6.1	Métrique d'erreur . . . . .	37
6.2	Étude du temps de calcul . . . . .	40
6.3	Simulation de l'algorithme de boucle fermée . . . . .	41
CHAPITRE 7 CONCLUSION . . . . .		43
7.1	Synthèse des travaux . . . . .	43
7.2	Limitations de la solution proposée et améliorations futures . . . . .	43
RÉFÉRENCES . . . . .		45
ANNEXES . . . . .		47

**LISTE DES TABLEAUX**

Tableau 6.1	Liste des variables utilisées dans l'algorithme de discrimination . . . . .	38
Tableau 6.2	Temps de calcul moyen pour chaque variable . . . . .	40

## LISTE DES FIGURES

Figure 1.1	Distance parcourue sur la Lune et sur Mars (NASA (2013)) . . . . .	2
Figure 1.2	Exemple de recalage . . . . .	5
Figure 1.3	Étapes principales de l’algorithme ICP . . . . .	6
Figure 1.4	Exemple illustrant le problème de minimum local . . . . .	6
Figure 3.1	Accumulation d’erreur de positionnement . . . . .	10
Figure 3.2	Trajectoire fermée avant et après correction . . . . .	11
Figure 3.3	Exemple de recalage - Conditions initiales . . . . .	15
Figure 3.4	Exemple de recalage - Résultats intermédiaires . . . . .	16
Figure 3.5	Exemple de recalage - Résultat final et positions réelles . . . . .	17
Figure 4.1	Représentation de la base de données selon deux variables . . . . .	20
Figure 4.2	Distribution de la distance entre chaque point et son plus proche voisin	21
Figure 4.3	Valeur de la fonction 4.2 en fonction du pourcentage de superposition .	22
Figure 4.4	Deux balayages ayant peu de relief (aucune erreur) . . . . .	23
Figure 4.5	Deux balayages ayant peu de relief (translation de 3 mètres) . . . . .	23
Figure 4.6	Exemple de balayage dans une région avec du relief . . . . .	24
Figure 4.7	Exemple de balayage dans une région plane . . . . .	25
Figure 4.8	Histogramme représentant la composante Z des normales . . . . .	25
Figure 4.9	Représentation de la distance d’Hausdorff (Rocchini, 2007) . . . . .	26
Figure 4.10	Transformation utilisée pour la technique « spin image » (Johnson, 1997)	27
Figure 4.11	Étapes de la génération de « spin image » à partir d’un nuage de points (Johnson, 1997) . . . . .	27
Figure 4.12	Exemple de points utilisés pour comparer les images . . . . .	29
Figure 4.13	Images produites et diagramme de dispersion pour une bonne corres- pondance . . . . .	29
Figure 4.14	Images produites et diagramme de dispersion pour une mauvaise cor- respondance . . . . .	30
Figure 4.15	Balayage utilisé pour l’exemple d’image de distance . . . . .	30
Figure 4.16	Exemple d’image de distance . . . . .	31
Figure 4.17	Bon recalage, pourcentage de points ayant un PPV à moins de 0.4m : 94.9% . . . . .	32
Figure 4.18	Mauvais recalage, pourcentage de points ayant un PPV à moins de 0.4m : 86.9% . . . . .	32
Figure 5.1	Terrain d’émulation de Mars (CSA, 2011) . . . . .	34

Figure 5.2	« Saddle Mountain » (CSA, 2011) . . . . .	34
Figure 5.3	Exemple de balayage près de « Saddle Mountain » (CSA, 2011) . . . .	35
Figure 5.4	Exemple de deux balayages près de la plaque de dalles Alpha (CSA, 2011) . . . . .	36
Figure 5.5	Montage expérimental du « Mobile Robotics Test-Bed » (CSA, 2011) .	36
Figure 6.1	APER des 10000 meilleures combinaisons de paramètres . . . . .	39
Figure 6.2	Pourcentage d'utilisation de chaque variable dans les 10000 meilleures combinaisons . . . . .	39
Figure 6.3	Exemple de recalage . . . . .	42
Figure A.1	Erreurs de translation en fonction de l'erreur de rotation . . . . .	47
Figure A.2	Région de superposition calculée par l'algorithme ICP . . . . .	48
Figure A.3	Région de superposition calculée par l'algorithme ICP . . . . .	49
Figure A.4	Région de superposition calculée selon la méthode de Chetverikov <i>et al.</i> (2005) . . . . .	50
Figure A.5	Nombre d'itérations . . . . .	51
Figure A.6	Distance d'Hausdorff . . . . .	52
Figure A.7	Normale des points de tout le balayage source - méthode 1 . . . . .	53
Figure A.8	Normale des points de tout le balayage cible - méthode 1 . . . . .	54
Figure A.9	Normale des points de la région de superposition du balayage source - méthode 1 . . . . .	55
Figure A.10	Normale des points de la région de superposition du balayage cible - méthode 1 . . . . .	56
Figure A.11	Normale des points de tout le balayage source - méthode 2 . . . . .	57
Figure A.12	Normale des points de tout le balayage cible - méthode 2 . . . . .	58
Figure A.13	Normale des points de la région de superposition du balayage source - méthode 2 . . . . .	59
Figure A.14	Normale des points de la région de superposition du balayage cible - méthode 2 . . . . .	60
Figure A.15	Coefficient de corrélation des « spin images » . . . . .	61
Figure A.16	Coefficient de confiance des « spin images » . . . . .	62
Figure A.17	Pourcentage de points finis des images de distance . . . . .	63
Figure A.18	Erreur quadratique moyenne des images de distance . . . . .	64
Figure A.19	MSSIM des images de distance . . . . .	65
Figure A.20	Pourcentage du nombre de plus proches voisins du balayage source . .	66
Figure A.21	Pourcentage du nombre de plus proches voisins du balayage cible . . .	67

## LISTE DES SIGLES ET ABRÉVIATIONS

APER	Apparent Error Rate
CRSNG	Conseil de Recherches en Sciences Naturelles et Génie
CSA	Canadian Space Agency
DGPS	Differential Global Positioning System
EMICP	Expectation Minimization Iterative Closest Point
GPS	Global Positioning System
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
LIDAR	Light Detection And Ranging
MET	Mars Emulation Terrain
MRT	Mobile Robotics Test-bed
NASA	National Aeronautics and Space Administration
PERLDB	Planetary Exploration Rover LIDAR Database
PPV	Plus Proche Voisin
RADAR	Radio Detection And Ranging
RTK	Real Time Kinematic
SVD	Singular Value Decomposition



## CHAPITRE 1

### INTRODUCTION

L’exploration extra-terrestrielle est très importante pour plusieurs domaines scientifiques. En effet, en étudiant les astres composant notre univers, en particulier ceux de notre système solaire, il est possible d’en apprendre plus sur la Terre et même sur l’origine de la vie. Par contre, cette exploration est évidemment très complexe et repousse les limites de nos technologies, notamment dans le domaine de la robotique. Depuis les années 70, plusieurs robots mobiles ont été envoyés vers la Lune et Mars. La figure 1.1 tirée du site de la NASA (NASA, 2013) représente les distances parcourues par les robots ayant atterri avec succès sur la Lune ou sur Mars. Sur la lune, l’autonomie est moins importante puisque le délai de communication avec la Terre est de seulement quelques secondes. Par contre, puisque Mars est beaucoup plus loin que la lune, le délai de communication devient de l’ordre de quelques dizaines de minutes et limite de beaucoup la téléopération des robots. Ils doivent donc avoir un degré d’autonomie plus important et ceci explique les plus courtes distances parcourues sur Mars. Un obstacle majeur pour l’autonomie de ces robots est le fait qu’ils évoluent dans un environnement pour lequel nous n’avons que très peu de connaissance. Les cartographies de la Lune et de Mars n’ont pas une assez bonne résolution pour représenter tous les obstacles nuisibles aux robots. Le problème de localisation et cartographie simultanées se pose alors. Il s’agit de pouvoir créer une carte de son environnement tout en pouvant s’y situer.

#### 1.1 Éléments de la problématique

Une technologie permettant aux robots d’étudier et de modéliser leur environnement est le LIDAR (« LIght Detection And Ranging »). Ceci consiste à émettre un faisceau laser et de capter la réflexion de celui-ci. Il est alors possible de calculer la distance de l’objet ayant réfléchi le laser en fonction du temps entre l’émission et la réception. En effectuant ceci pour différents angles d’azimut et d’élévation, il est possible d’avoir un nuage de points en coordonnées sphériques et éventuellement en coordonnées cartésiennes à la suite d’une simple transformation. Cette technologie permet de représenter fidèlement l’environnement immédiat du robot mais perd de la résolution pour des distances plus éloignées. De plus, un objet relativement petit peut cacher une grande partie du champ de vision du LIDAR si celui-ci est très près du robot. Il est donc inévitable de se déplacer pour effectuer un balayage à un autre endroit et de faire la correspondance entre ces deux informations. La consolidation

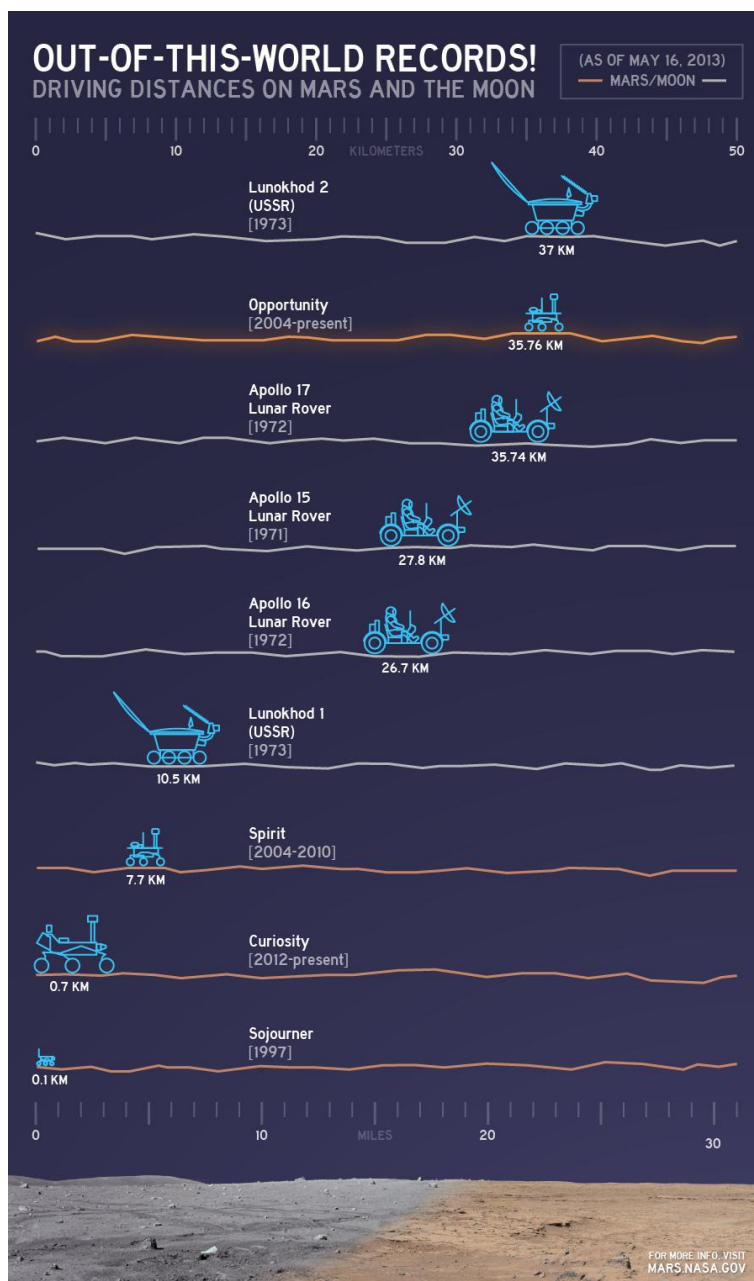


Figure 1.1 Distance parcourue sur la Lune et sur Mars (NASA (2013))

de deux balayages se fait par la technique de recalage.

Le recalage est le fait d'établir la correspondance entre deux images représentant en partie un même objet ou dans notre cas une même surface. La figure 1.2 démontre un exemple de recalage de deux images. Ces deux images représentent chacune une partie de la surface autour d'un cratère. Le recalage de ces images permet alors d'avoir une carte mieux définie du cratère et de ses alentours.

Pour effectuer le recalage, l'algorithme le plus utilisé est l'algorithme ICP pour « Iterative Closest Point ». Comme son nom l'indique, c'est un algorithme itératif permettant de trouver la relation entre les images donnant la plus petite distance entre chacun des points. Comme l'illustre le schéma 1.3, les étapes principales de l'algorithme sont la sélection de points du balayage source, le calcul du plus proche voisin de chacun de ces points dans le balayage cible et l'application d'une transformation minimisant la distance entre ces paires de points.

L'algorithme ICP de référence est basé sur les travaux de Besl et McKay (1992). Par contre, plusieurs auteurs ont repris cet algorithme et l'ont modifié pour l'adapter à leur besoin en fonction de l'application. Dans le cadre de ce projet, l'algorithme utilisé a été développé par Sébastien Gemme de l'Agence Spatiale Canadienne. Certaines différences s'y retrouvent par rapport à l'ICP de base. Premièrement, le recalage se fait en deux parties. Un premier recalage se fait avec des paramètres moins limitatifs permettant d'obtenir un résultat grossier rapidement. Ensuite, ce résultat est amélioré en faisant un autre recalage avec des paramètres plus restrictifs. L'algorithme a aussi été amélioré en ajoutant la normale à la surface dans la recherche des points voisins. En effet, au lieu de faire une recherche dans un espace en trois dimensions, la correspondance entre deux points est établie en fonction de la normale à la surface en plus des coordonnées cartésiennes. Ceci permet de faire une meilleure correspondance entre les points de deux balayages.

Malgré ceci, le recalage connaît une limitation importante. Celui-ci est très sensible aux conditions initiales et risque de converger vers un minimum local si l'estimation de la transformation reliant deux balayages est relativement loin de la réalité. Ce problème est illustré à la figure 1.4. Imaginons que nous voulons trouver le minimum de la courbe illustrée. La technique classique est de se déplacer dans la direction de la pente à partir de la condition initiale. Lorsqu'un minimum est atteint, la pente de chaque côté du point ramène toujours au même point qui est considéré comme le minimum. Une façon intuitive d'illustrer le problème est en s'imaginant une bille qui roule sur la courbe. Celle-ci va se retrouver dans un des trois minimums indiqués sur le graphique en fonction de la condition initiale. Il est évident que si cette condition est à l'extérieur des lignes pointillées, nous n'obtiendrons pas le minimum recherché.

C'est exactement le même problème rencontré lors de l'algorithme de recalage. Celui-ci effectue de petites transformations à chaque itération. Or, il est possible d'atteindre un point où une petite transformation va augmenter la distance entre les points sélectionnés même si les balayages ne sont pas bien recalés. L'algorithme en conclut qu'il a atteint le minimum, mais celui-ci ne représente pas nécessairement le bon recalage. Présentement, il n'y a aucun moyen de quantifier le résultat du recalage et celui-ci est toujours considéré comme bon. Ceci s'avère être un problème important pour la cartographie puisque des erreurs importantes

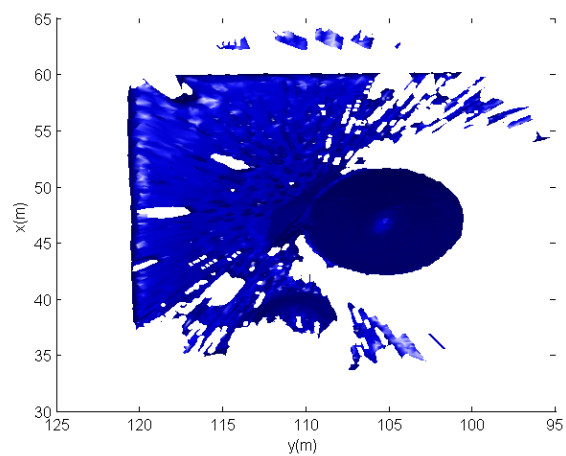
peuvent ainsi être introduites dans la carte globale.

## **1.2 Objectifs de recherche**

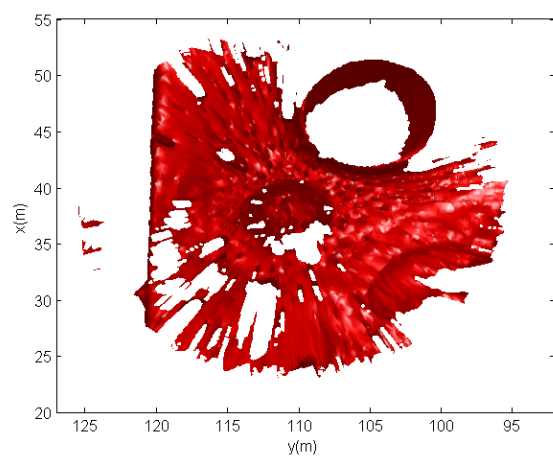
L'objectif principal de ce projet de recherche consiste à déterminer une métrique d'erreur permettant de quantifier le résultat d'un recalage et ainsi rejeter les résultats erronés. Le résultat recherché est donc une valeur de 0 à 1 représentant la probabilité qu'un recalage ait convergé vers le minimum absolu. De plus, cette information pourrait être utilisée dans un algorithme de répartition d'erreur pour redistribuer l'erreur accumulée tout au long de la trajectoire d'un robot mobile lorsque celui-ci revient à un endroit déjà visité.

## **1.3 Plan du mémoire**

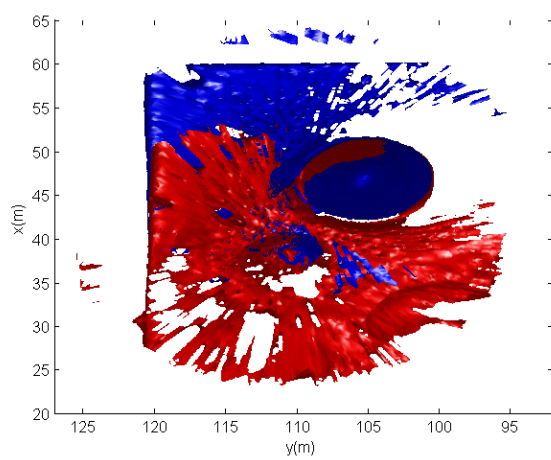
Les grandes lignes du travail fait pour rencontrer les objectifs sont présentés dans le présent document. Tout d'abord, une brève revue de littérature sera faite pour expliquer l'état des recherches dans le domaine de localisation et cartographie simultanées. Ensuite, l'algorithme de boucle fermée utilisé pour distribuer l'erreur sera présenté. Ceci permettra d'introduire le chapitre suivant qui correspond au coeur du projet, soit la détermination d'une métrique permettant de discriminer les bons des mauvais recalages. Une brève description des données utilisées sera aussi faite avant de présenter les résultats obtenus. Finalement, une conclusion permettra d'établir l'état des travaux et de proposer des pistes de continuation.



(a) Image 1



(b) Image 2



(c) Images 1 et 2 recalées

Figure 1.2 Exemple de recalage

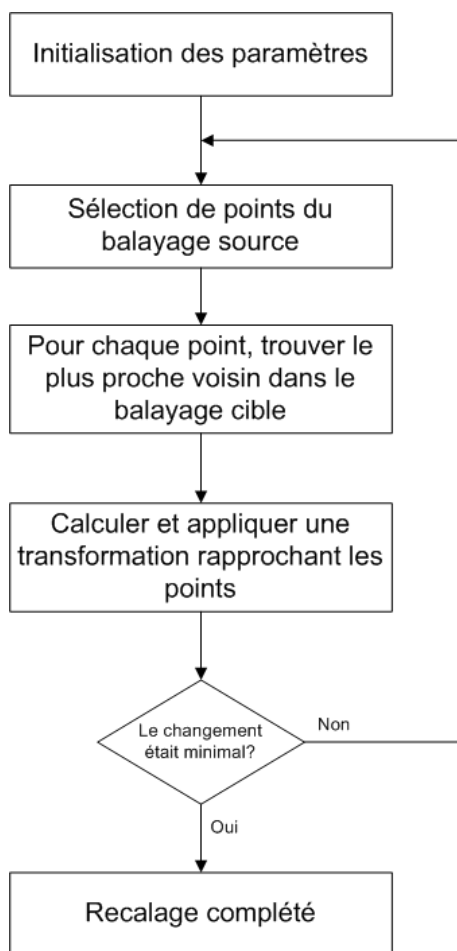


Figure 1.3 Étapes principales de l'algorithme ICP

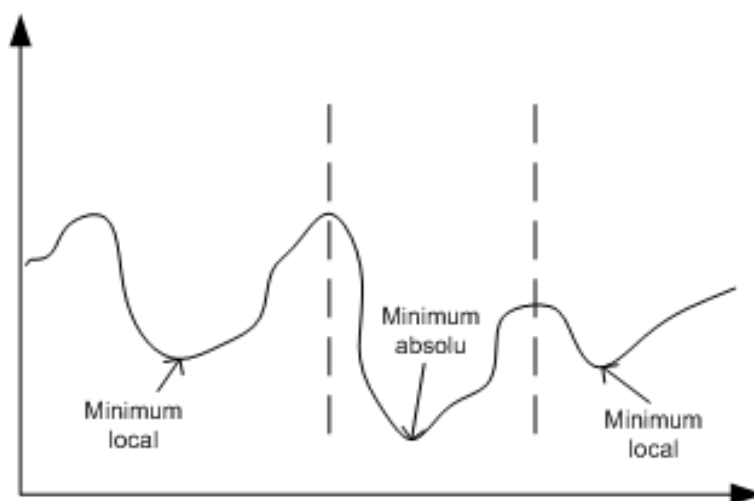


Figure 1.4 Exemple illustrant le problème de minimum local

## CHAPITRE 2

### REVUE DE LITTÉRATURE

Nous présenterons ici un bref aperçu de la littérature sur le sujet étudié. Cette revue est séparée en trois sections. La première partie rassemble les travaux faits pour le recalage de deux images ensemble. Ensuite, les recherches portant sur la minimisation de l'erreur lorsque nous avons plusieurs images à recaler entre elles seront présentées. Finalement, les études faites plus particulièrement pour des applications de localisation et cartographie simultanées en robotique mobile seront listées.

#### 2.1 Algorithme ICP et variantes

Le recalage d'image est un sujet très étudié depuis de nombreuses années. Il faut remonter jusqu'à plus de vingt ans pour voir l'apparition d'un des articles les plus cités dans ce domaine, soit celui de Chen et Medioni (1991). Les auteurs proposent une technique de recalage en minimisant la distance entre un point et une surface. Très peu longtemps après, Besl et McKay (1992) proposent l'algorithme ICP standard décrit brièvement dans l'introduction. Ils prouvent que cet algorithme converge de façon monotone vers le minimum local le plus près.

Par la suite, plusieurs auteurs ont repris cet algorithme pour l'améliorer et l'adapter à leur problème. Ainsi, parmi les techniques développées, notons l'EMICP (Liu, 2006), (Liu, 2010) qui associe l'ICP avec une technique d'espérance et de minimisation (EM). Cette approche permet d'ajouter un calcul probabiliste pour déterminer la correspondance entre les points de deux recalages.

Sharp *et al.* (2002) se sont aussi concentrés sur la correspondance entre les points des deux images à recaler. Ils ont étudié plusieurs quantités invariantes par rapport à des translations et rotations des images. Ces quantités sont entre autres la courbure des surfaces, les moments invariants, les harmoniques sphériques, etc. Barnea et Filin (2008) ont quant à eux développé une technique se basant sur la distance des points dans un même nuage de points. Cette valeur est elle aussi invariante par rapport à une transformation et permet de faire une meilleure correspondance entre les deux images.

Notons aussi les travaux de Choe et Medioni (2007) qui font du recalage d'image presque planaires. Le recalage est d'abord estimé en supposant les images planaires. Ensuite, une reconstruction en trois dimensions de la surface est effectuée et un autre recalage entre ces

nouvelles images permet d'obtenir un résultat plus précis. Aussi, Du *et al.* (2010) ont modifié l'algorithme ICP pour l'utiliser dans le cas affine. La matrice de transformation est alors décomposée en trois matrices par la méthode SVD. Celles-ci représentent la réflexion, la rotation et l'homothétie.

Les variantes sont tellement nombreuses que des chercheurs se sont occupés à comparer celles-ci. Rusinkiewicz et Levoy (2001) ont séparé l'algorithme selon six critères et ont comparé plusieurs variantes en fonction de ces critères. Ceux-ci sont l'échantillonnage des points, la correspondance des points, le poids associé à chaque correspondance, le critère de rejet des paires de points, la métrique d'erreur et la technique de minimisation de l'erreur.

Dans le cadre de ce projet, l'algorithme développé par Gemme *et al.* (2012) a été utilisé. Les principales différences de cet algorithme avec l'ICP standard sont discutées au chapitre 1.

## 2.2 Recalage de plusieurs surfaces

Dans la majorité des situations, le nombre de surfaces à recaler ne se limite pas seulement à deux. Eggert *et al.* (1996) ont présenté une technique pour distribuer l'erreur dans le cas où plusieurs surfaces sont disponibles. Ils relient chaque surface à des ressorts virtuels et le problème devient alors un simple problème de dynamique où les ressorts auront tendance à converger vers une position représentant un minimum d'énergie. Ceci permet de redistribuer l'erreur sur l'ensemble des recalages et d'avoir une carte globale plus continue.

Neugebauer (1997) quant à lui, utilise une méthode plus directe qui consiste à effectuer le recalage de toutes les surfaces en même temps. Il utilise la méthode de Levenberg-Marquardt pour résoudre le problème des moindres carrés et converger vers une solution où toutes les surfaces seront recalées entre elles. Wang *et al.* (2008) font eux aussi directement le recalage de plusieurs nuages de points en même temps. Par contre, ils modélisent ces nuages de points par des Gaussiennes et utilisent le critère de Jenson-Shannon pour minimiser l'erreur globale.

Dans ses travaux, Pulli (1999) effectue le recalage de surfaces deux par deux pour obtenir un ensemble de contraintes. Ces contraintes sont alors mises en commun pour obtenir un ensemble représentant toutes les surfaces. Il est alors possible de réduire la mémoire et la puissance de calcul nécessaire puisque les surfaces n'ont pas à être gardées en mémoire par la suite.

## 2.3 Applications spécifiques en robotique mobile

Il n'y a pas seulement en robotique mobile où le recalage est important. En effet, cette technique est très utilisée par exemple en imagerie médicale et en informatique pour la recon-



naissance d'objets. Ces applications utilisent des images ou surfaces très différentes de celles rencontrées en exploration spatiale où celles-ci sont très irrégulières et n'ont pas de marqueurs spécifiques permettant d'aider au recalage. C'est pourquoi il est intéressant de souligner les travaux de Surmann *et al.* (2003), Lingemann *et al.* (2005), Ohno *et al.* (2006), Nüchter *et al.* (2007) et Viejo et Cazorla (2007) qui utilisent des technique semblables à celles présentées plus haut mais qui sont mieux adaptées au genre de surfaces rencontrées lors du problème de localisation et cartographie simultanées.

Il est important de noter que dans les travaux présentées, aucun ne développe une métrique d'erreur permettant de qualifier le résultat d'un recalage à l'exception de ceux de Cole et Newman (2006). Ils utilisent les histogrammes représentant la distance entre les points à chaque itération pour distinguer un bon d'un mauvais résultat. Par contre, ce travail est fait avec des balayages effectués dans un environnement urbain où plusieurs facteurs aident grandement au recalage. En effet, une étape de pré-traitement permet de détecter des arêtes ou des formes particulières permettant de faciliter la correspondance entre les surfaces.

## CHAPITRE 3

### ALGORITHME DE BOUCLE FERMÉE

Un problème majeur pour l'autonomie des robots mobiles est l'accumulation d'erreurs qui entraîne la déviation de la trajectoire. Imaginons un robot qui tente de suivre une trajectoire rectiligne. Celui-ci prend un balayage de son environnement à tous les 5 mètres pour calculer sa nouvelle position en fonction de son dernier estimé. Par contre, supposons que l'estimation de la pose du robot est erronée de 0.1 degrés en rotation par rapport à l'axe Z (angle de lacet) à chaque étape. Cette erreur aura peu d'impact sur une courte distance, mais elle va rapidement devenir très importante. La figure 3.1 représente cette situation hypothétique. Après 50 mètres, l'erreur est de seulement 0.4 mètre tandis qu'elle est d'un peu plus que 1.6 mètres pour une trajectoire de 100 mètres.

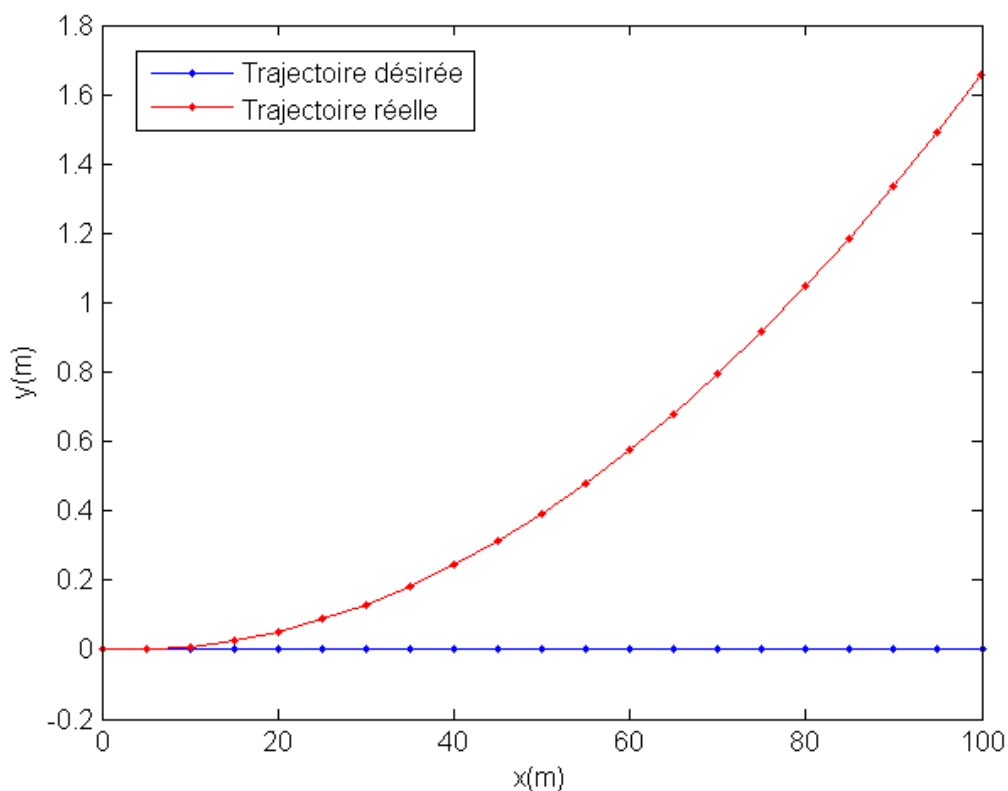


Figure 3.1 Accumulation d'erreur de positionnement

Pour une trajectoire rectiligne utilisée dans cet exemple, il est impossible pour le robot de

corriger cette accumulation d'erreur si celui-ci n'obtient aucune information supplémentaire. Par contre, si sa trajectoire n'est pas rectiligne et qu'il repasse au même endroit plus d'une fois, il est possible de distribuer l'erreur totale sur toute la trajectoire représentant la boucle fermée pour obtenir une carte globale plus précise. La figure 3.2 représente bien cette situation. La première image représente une trajectoire circulaire où le robot a effectué 7 balayages notés de  $N_1$  à  $N_7$ . Le premier et le dernier balayage ne se superposent pas selon l'estimation du robot, mais ils ont une région commune en réalité. Chaque ressort représente un lien entre deux balayages. Ce lien est en fait le résultat de l'algorithme de recalage. Le but est de corriger cette trajectoire comme le démontre la figure 3.2(b). Le lien entre les balayages  $N_1$  et  $N_7$  a permis de corriger l'erreur accumulée tout au long de la trajectoire et d'obtenir une carte plus précise.

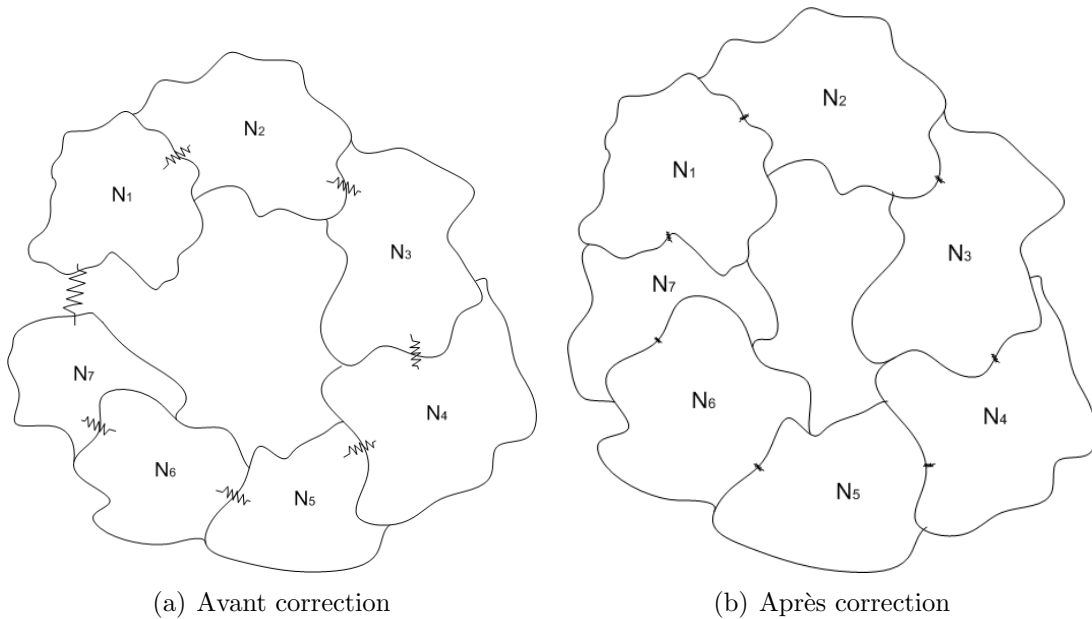


Figure 3.2 Trajectoire fermée avant et après correction

La technique utilisée pour faire la correction de l'erreur accumulée est celle proposée par Eggert *et al.* (1996) où des ressorts virtuels sont attachés entre chaque balayage. Pour utiliser cette méthode, il faut tout d'abord détecter quand le robot visite un endroit déjà connu. Autrement, les liens entre les balayages ne formeraient pas une boucle fermée et il serait donc impossible de corriger l'accumulation d'erreur. Deuxièmement, il faut déterminer le nombre, l'emplacement et la force des ressorts entre chaque paire de balayage. Finalement, un algorithme doit permettre de distribuer l'erreur sur l'ensemble de la boucle fermée selon les contraintes des ressorts.

La détection de boucle fermée ne fait pas partie de l'objectif de ce projet. Il est donc pris

pour acquis qu'un autre algorithme permet de détecter cette boucle ou qu'une intervention humaine est nécessaire pour cette partie. Les sections suivantes décriront comment les autres parties du processus ont été effectuées.

### 3.1 Nombre et emplacement des ressorts

Le nombre de ressorts utilisés a été fixé à trois pour éviter d'avoir un système sous-défini. Pour la même raison, les points choisis pour fixer les ressorts ne doivent évidemment pas être colinéaires. Pour choisir ceux-ci, une première sélection est faite en mesurant la distance de chaque point avec son plus proche voisin dans l'autre ensemble. La première paire de points sélectionnée est celle dont les points sont le plus près l'un de l'autre. Ensuite, tous ceux séparés de moins de 150% de la distance séparant cette première paire de points sont sélectionnés. Cette valeur a été fixée de manière à obtenir le plus grand sous-ensemble possible composé seulement des meilleurs correspondances. Par contre, si moins de 20 paires de points respectent ce critère, les 20 premières sont sélectionnées pour éviter d'avoir des points colinéaires. Finalement, des vecteurs sont créés entre ces points et le premier point choisi ( $\vec{v}_{21}$  et  $\vec{v}_{31}$ ). Les deux points donnant le plus grand produit vectoriel sont ceux qui seront utilisés pour fixer les ressorts.

### 3.2 Algorithme de répartition de l'erreur

Une fois les ressorts fixés, la distribution de l'erreur se fait en résolvant le système dynamique composé des balayages et des ressorts. Ceci est fait itérativement avec un intervalle de temps  $\Delta t$  assez petit. Ceci permet de calculer les déplacements d'un balayage par rapport aux autres en supposant ceux-ci fixes. Il suffit donc de faire la somme des forces et couples appliquées à un balayage par les ressorts. Ensuite, les équations 3.1, 3.2, 3.3 et 3.4 permettent d'obtenir un déplacement linéaire et angulaire. Les vitesses linéaires et angulaires initiales ( $\vec{v}_0$  et  $\vec{\omega}_0$ ) sont considérées comme nulles à chaque itération. Cette approximation ne change pas le résultat puisque le système va néanmoins converger vers le minimum. Par contre, l'algorithme prendra plus d'itérations. Une fois les déplacements calculés pour chacun des balayages, ils sont appliqués et l'algorithme itère comme ceci jusqu'à ce que la somme de la norme de tous les déplacements soit plus petite qu'une valeur seuil.

$$\Sigma \vec{F} = m\vec{a} \quad (3.1)$$

$$\vec{d} = \vec{d}_0 + \vec{v}_0 \Delta t + \frac{\vec{a} \Delta t^2}{2} \quad (3.2)$$

$$\Sigma \vec{\tau} = I \vec{\alpha} \quad (3.3)$$

$$\vec{\theta} = \vec{\theta}_0 + \vec{\omega}_0 \Delta t + \frac{\vec{\alpha} \Delta t^2}{2} \quad (3.4)$$

Pour éviter d'avoir une divergence causée par l'approximation des balayages fixes, l'intervalle de temps  $\Delta t$  de chaque itération est dynamique. Si un déplacement est supérieur à celui de la dernière itération, celui-ci n'est pas appliquée et  $\Delta t$  est réduit de 80%. Ceci permet d'avoir un intervalle de temps plus grand au début de l'algorithme lorsque les déplacements sont plus importants et celui-ci diminue au fur et à mesure que l'algorithme itère. La valeur de 80% correspond au compromis entre une trop petite valeur qui diminuerait trop rapidement  $\Delta t$  versus une plus grande valeur qui demanderait plus d'itération pour réduire le critère. De cette façon, il est possible d'avoir un résultat précis sans avoir un nombre excessif d'itérations.

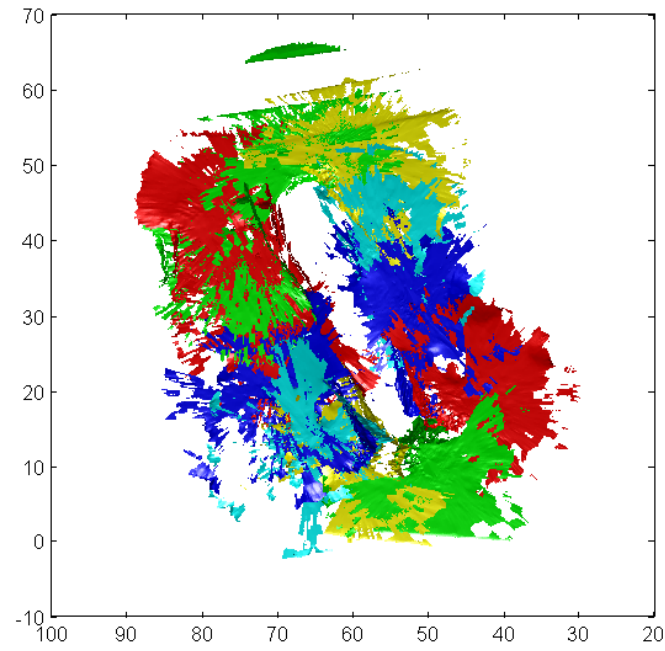
Un exemple du résultat que cet algorithme peut donner est représenté aux figures 3.3, 3.4 et 3.5. Imaginons un parcours circulaire d'un robot où 12 balayages ont été faits. La figure 3.3 représente cette trajectoire où une erreur sur la position et l'orientation de chacun des balayages a été ajoutée. Ensuite, étudions d'abord le cas où la correspondance entre les balayages est connue. Ceci permet de fixer chaque ressort entre deux points représentant le même espace physique. La figure 3.4 donne quant à elle la position des balayages après 2 et 5 itérations de l'algorithme et le résultat final ainsi que les positions réelles se retrouvent à la figure 3.5. L'amplitude de l'erreur moyenne obtenue est de 0.19 mètre et de 0.51 radians pour chaque balayage.

### 3.3 Force des ressorts

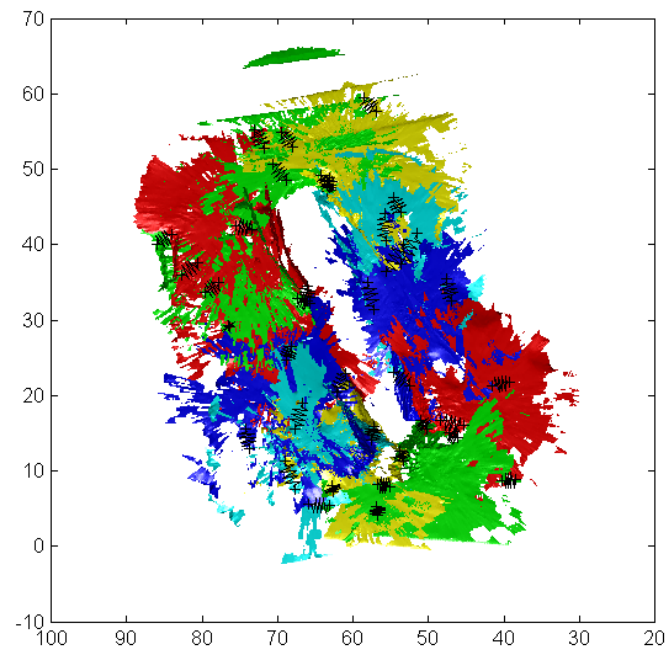
Il serait avantageux d'utiliser une constante de raideur différente pour chaque ressort en fonction de la confiance au résultat du recalage. Dans l'exemple précédent, les ressorts ont tous une constante de  $1 \text{ N/m}$ . Par contre, les recalages n'ont pas forcément tous donné la même précision. Certains auront convergé vers le minimum absolu et permettent de placer les ressorts aux endroits idéaux. Par contre, d'autres vont introduire une erreur s'ils convergent vers un minimum local. Le cas idéal serait de détecter les mauvais recalages et ainsi de rejeter la correspondance établie entre les deux balayages concernés. L'algorithme pourrait alors utiliser seulement les résultats de recalage fiables pour redistribuer l'erreur. Par contre, il est très difficile de quantifier le résultat de l'algorithme de recalage. Il n'y a pas d'information évidente donnant la validité du résultat.

Le chapitre suivant porte donc sur la recherche d'une métrique pouvant indiquer la probabilité que le résultat d'un recalage représente le minimum absolu. Cette valeur pourrait

alors être directement utilisée pour fixer la constante de raideur des ressorts dans l'algorithme de distribution d'erreur. Ainsi, un ressort ayant une constante de raideur plus élevée représenteraient des liens plus rigide et les balayages qui y sont attachés auront moins tendance à bouger. À l'inverse, un recalage erroné engendrerait une constante faible et les balayages pourront être déplacés pour corriger cette erreur.

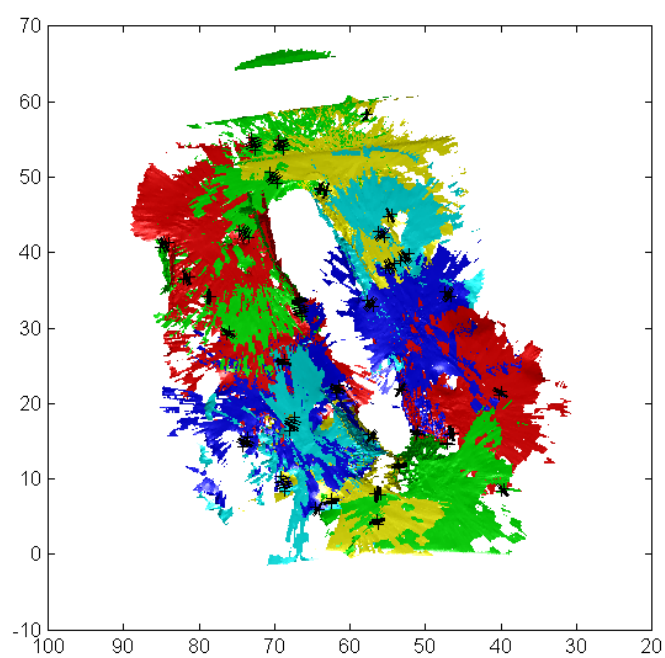


(a) Conditions initiales

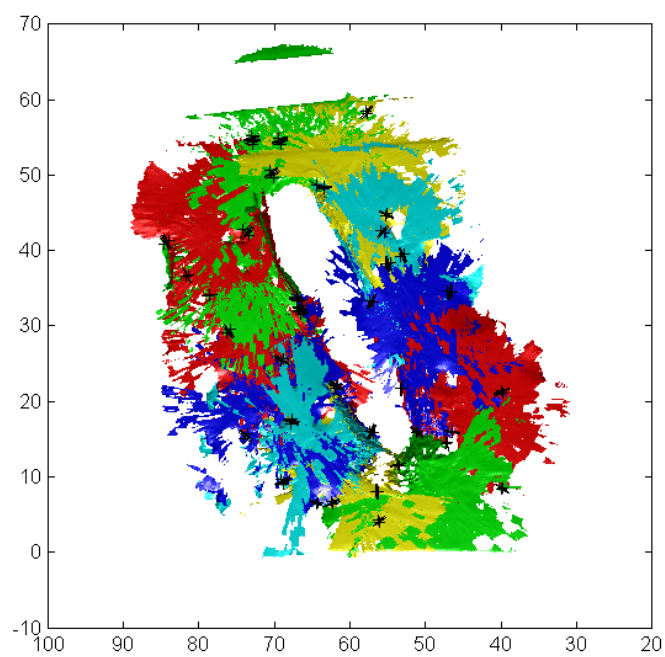


(b) Ressorts fixés

Figure 3.3 Exemple de recalage - Conditions initiales



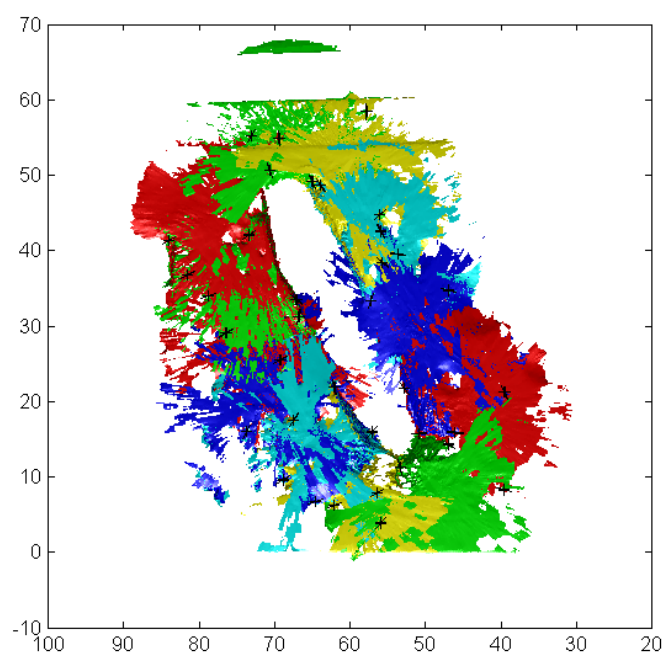
(a) Itération 2



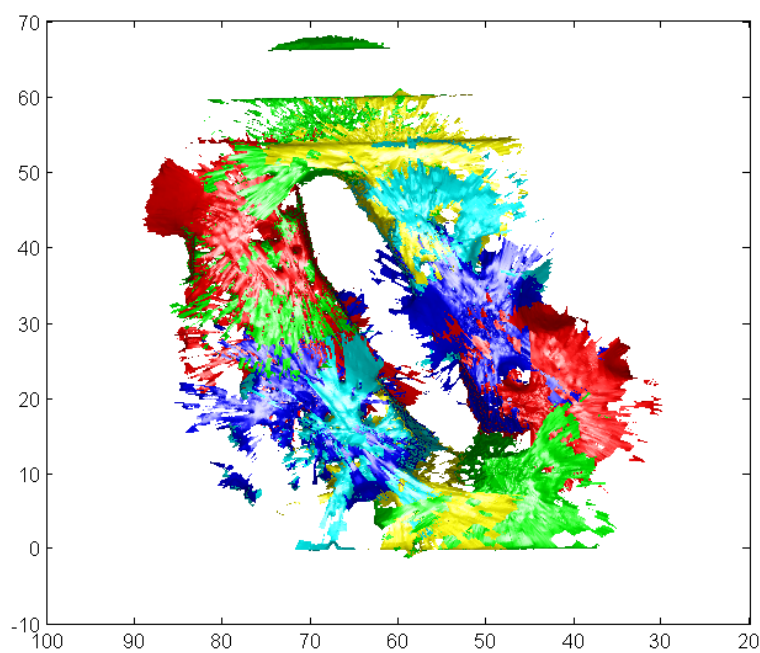
(b) Itération 5

Figure 3.4 Exemple de recalage - Résultats intermédiaires





(a) Résultat final



(b) Positions réelles

Figure 3.5 Exemple de recalage - Résultat final et positions réelles

## CHAPITRE 4

### DÉTERMINATION D'UNE MÉTRIQUE

L'objectif est de trouver une caractéristique du résultat d'un recalage pour détecter si celui-ci a convergé vers le minimum absolu. Par exemple, un recalage aurait plus de chance de converger plus la région de superposition entre les balayages est grande. Cette simple variable pourrait donc donner une indication sur la validité du résultat. Cependant, la superposition des balayages ne permet pas de bien différencier un bon d'un mauvais résultat. Il faut donc trouver une variable ou plus vraisemblablement une combinaison de variables permettant d'effectuer cette différenciation. Les étapes suivantes ont permis d'obtenir une métrique s'approchant de cet objectif.

Tout d'abord, une base de données de résultats de recalages a été construite pour effectuer les tests. Chaque résultat est alors modélisé par un ensemble de variables le caractérisant. Plusieurs facteurs ont été testés et 19 de ceux-ci ont été retenus. Ensuite, un algorithme de discrimination a été appliqué avec chacune des combinaisons possibles de ces 19 variables pour finalement obtenir l'ensemble des caractéristiques permettant de distinguer un bon d'un mauvais recalage.

#### 4.1 Création de la base de données

À partir des 102 nuages de points pris sur le terrain d'émulation de Mars (voir chapitre 5), 169 paires ont été sélectionnées pour tester l'algorithme de recalage. Ces paires correspondent à toutes les paires de nuages distants de moins de 10 mètres auxquels ont été enlevés ceux qui ont une trop petite région de superposition. Pour chacune de ces paires de balayages, une erreur variant de -25 à +25 degrés pour l'angle de lacet ainsi que de 0 à 5 mètres en translation a été ajoutée au balayage source avant d'appliquer l'algorithme ICP. Ceci a donné un résultat de recalage pour 4203 situations. Pour chaque recalage, la position corrigée est comparée à la position réelle du balayage. Si la distance est inférieure à 0.2 mètre et 0.05 radian ( $\approx 2.86^\circ$ ), le résultat est considéré comme bon. Au contraire, si ces valeurs sont de plus de 1 mètre ou 0.2 radian ( $\approx 11.46^\circ$ ), le résultat est étiqueté comme mauvais. Ainsi, le résultat peut être ni bon, ni mauvais puisque seulement les extrêmes sont considérés pour la détermination de la métrique. Ceci permet de faire une délimitation nette entre un bon et un mauvais résultat pour mieux détecter les caractéristiques propres à chaque groupe.

## 4.2 Algorithme de discrimination

La technique utilisée pour qualifier le résultat d'un recalage est l'analyse discriminante. À partir de la base de données, nous avons deux groupes bien distincts représentant les bons et les mauvais résultats. Chaque test est représenté dans un espace à N dimensions correspondant aux N variables permettant de mieux décrire le résultat d'un recalage. Pour chaque groupe, la moyenne ( $\mu$ ) et la covariance ( $\Sigma$ ) des N variables sont calculés et permettent d'obtenir une fonction de distribution gaussienne multivariable. Lorsqu'un nouveau recalage est effectué, la probabilité d'appartenance à chacun des groupes est calculée selon la formule 4.1 et la plus haute valeur indique alors à quel groupe le nouveau recalage appartient.

$$f_x = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (4.1)$$

Par exemple, imaginons que nous avons un espace en deux dimensions (N=2). La base de données pourrait alors être représentée dans un plan comme à la figure 4.1 où les points bleus sont les bons résultats et les points rouges les mauvais. Les ellipses de même couleur représentent l'espace d'écart-type de 1 (« 1-sigma ellipse »). La plus grande région bleue correspond à la région où les nouveaux points seront considérés comme des résultats valides et les points à l'extérieur seront invalides. L'exemple de la figure 4.1 contient des valeurs fictives, mais représente un cas probable où les ellipses des deux groupes se superposent ce qui implique inévitablement des mauvaises classifications. L'index de performance de l'algorithme s'appelle le taux d'erreur apparent (APER « APparant Error Rate ») et correspond au taux du nombre de mauvaises classifications sur le nombre total de points. Dans cet exemple, il y a 28 points mal classifiés (points rouges dans la région bleue et points bleus à l'extérieur) sur un total de 200 points. Ceci indique donc qu'un nouveau recalage aurait 14% de chance d'être mal classifié selon ces critères. Évidemment, plus les deux groupes sont distincts, plus le taux d'erreur apparent diminue et la classification se fait plus adéquatement. La difficulté réside donc dans le choix du nombre et de la nature des variables permettant d'obtenir des ellipses les plus distinctes entre les bons et les mauvais recalages. La section suivante énumère les principales variables testées et donne une brève description de chacune d'elles.

## 4.3 Variables

Chaque variable a été testée indépendamment pour voir si elle permet de différencier les deux groupes définis plus haut. En affichant la dite variable de tous les recalages sélectionnés en fonction de leur erreur, il est facile de détecter visuellement si la variable donne une bonne indication de la validité de chaque recalage. Les graphiques de chacune des variables

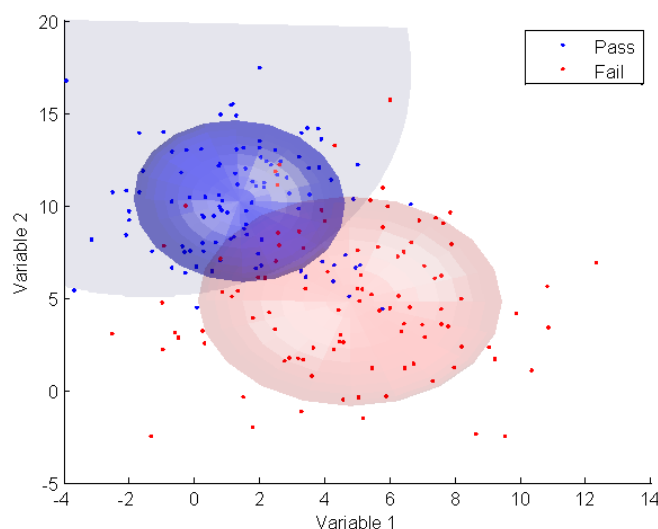


Figure 4.1 Représentation de la base de données selon deux variables

en fonction de l'erreur de translation sont réunis dans l'Annexe 1.

Pour chaque résultat de recalage, les caractéristiques suivantes ont été comparées avec la base de données pour trouver la meilleure façon de différencier les résultats valides des autres.

- Pourcentage de superposition
- Le nombre d'itérations
- La différence entre les distributions des normales
- L'amplitude de la correction effectuée
- La distance de chaque point avec son plus proche voisin

#### 4.3.1 Pourcentage de superposition

Le pourcentage de superposition est très important dans l'algorithme de recalage. En effet, plus la surface recouverte par les deux balayages est grande, plus il y a de chances que l'algorithme converge vers la bonne solution. Inversement, si la région commune aux deux surfaces est très petite, il est très improbable d'obtenir une bonne solution. Le pourcentage de la surface qui est commune entre les deux balayages donne une indication de la probabilité d'avoir un bon recalage. Deux techniques ont été testées pour calculer cette valeur.

Premièrement, la distance de chaque point d'un balayage avec son plus proche voisin dans l'autre balayage est calculée. Ensuite, une valeur seuil permet de déterminer quels points font partie de la région de superposition. La figure 4.2 représente une partie de l'histogramme obtenu ainsi. On y voit très bien qu'une grande majorité des points ont un plus proche voisin à 0.25 mètre ou moins. Cette valeur a donc été conservée comme valeur de seuil pour ce

test. Dans cet exemple-ci, 9929 points sur 13940 répondent à ce critère, le pourcentage de superposition est donc de 71.2%.

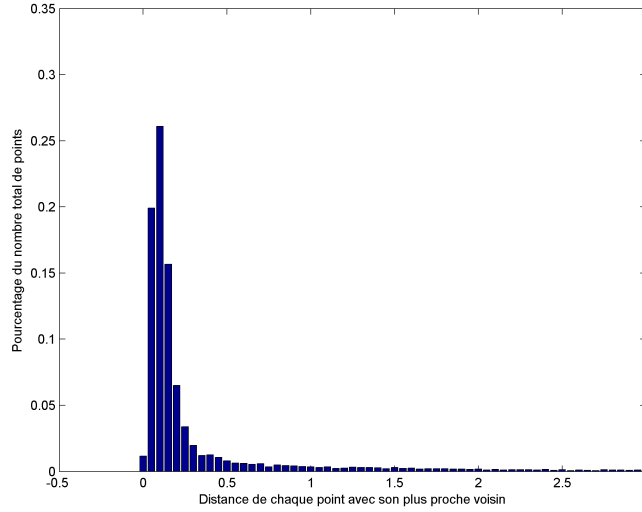


Figure 4.2 Distribution de la distance entre chaque point et son plus proche voisin

L'autre technique utilisée pour calculer le pourcentage de superposition est celle présentée par Chetverikov *et al.* (2005). Celui-ci propose de minimiser la fonction objectif 4.2 représentant le ratio entre l'erreur quadratique moyenne et le pourcentage de points dans la région de superposition. Le paramètre  $\lambda$  permet de fixer l'importance du pourcentage de superposition dans ce ratio. Celui-ci a été fixé à 2 comme dans les travaux de Chetverikov *et al.* (2005).

$$\psi(\xi) = \frac{e(\xi)}{\xi^{1+\lambda}} \quad (4.2)$$

En calculant ce rapport pour différentes valeurs de  $\xi$ , le minimum obtenu correspond au pourcentage de superposition recherché. En reprenant le même exemple que précédemment, nous obtenons le graphique de la figure 4.3 représentant la fonction à minimiser. Le minimum se situe à 0.73, ce qui donne un pourcentage de superposition relativement proche de celui obtenu par l'autre méthode (73% et 71.2%).

#### 4.3.2 Nombre d'itérations

L'algorithme de recalage s'effectue en deux parties. Premièrement, un recalage grossier permet de d'obtenir une bonne approximation de la solution. Ensuite, un recalage plus restrictif permet de trouver la réponse la plus précise possible. La première étape s'effectue en un maximum de 50 itérations et la deuxième en 5 itérations ou moins. Le nombre maximal

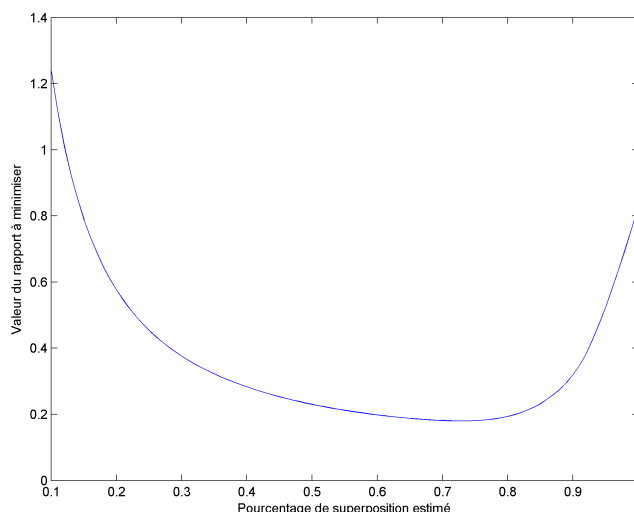


Figure 4.3 Valeur de la fonction 4.2 en fonction du pourcentage de superposition

d'itérations est donc de 55. Cette valeur à elle seule ne permet pas de différencier un bon d'un mauvais recalage puisque si l'algorithme termine en moins de 55 itérations, ceci signifie seulement qu'il a atteint un minimum local et rien ne garantit que ce minimum est la solution recherchée. Inversement, un algorithme n'ayant pas atteint le critère de convergence avant 55 itérations peut néanmoins être très près du minimum global. Malgré ceci, il est possible que le nombre d'itérations soit pertinent pour valider le résultat du recalage en l'utilisant avec d'autres variables. Par exemple, un pourcentage de superposition faible pourrait expliquer qu'un algorithme prenne plus d'itérations pour converger ou bien un faible relief expliquerait un nombre d'itérations petit.

### 4.3.3 Relief des balayages

Après plusieurs tests de l'algorithme de recalage, il est devenu évident que le relief du terrain a un grand impact sur les performances de l'algorithme. Celui-ci converge beaucoup plus souvent vers un minimum local lorsque le terrain est très plat que lorsqu'il y a plus de relief. Même pour un être humain, le recalage peut devenir très difficile dans cette situation. Prenons par exemple la figure 4.4 qui représente deux balayages effectués près de la plaque de dalles alpha (région très plane). Les deux balayages ayant très peu de relief, il est pratiquement impossible de dire avec certitude qu'ils sont bien superposés. La figure 4.5 représente les mêmes balayages, mais une erreur de translation de 3 mètres a été ajoutée à un balayage. En comparant les deux figures, on ne saurait dire laquelle des deux est le résultat d'un recalage erroné.

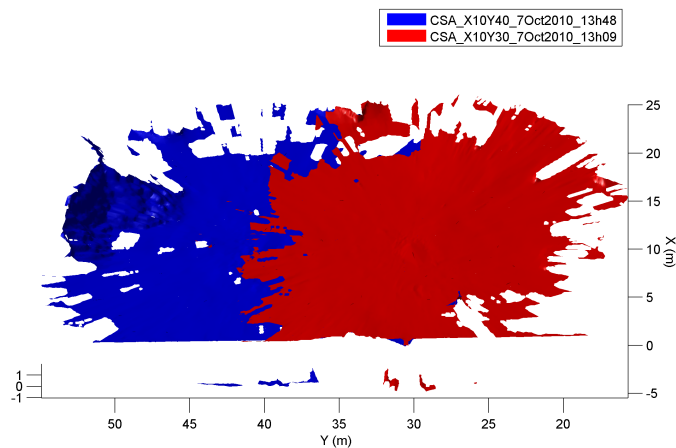


Figure 4.4 Deux balayages ayant peu de relief (aucune erreur)

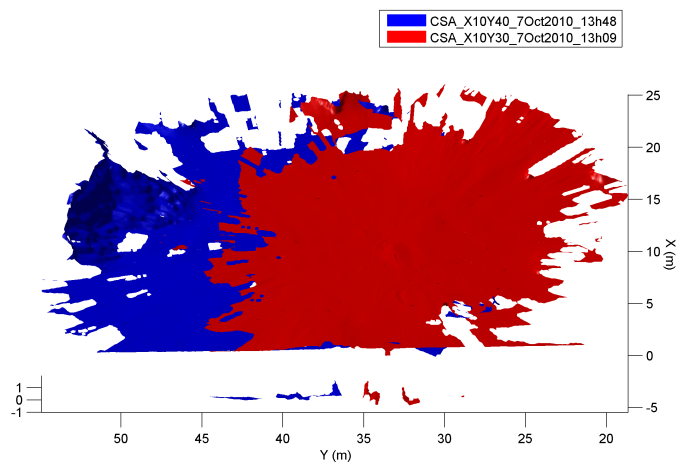


Figure 4.5 Deux balayages ayant peu de relief (translation de 3 mètres)

Une information qui peut indiquer le niveau de relief dans un nuage de point et qui a l'avantage d'être déjà calculée est la normale de la surface en chaque point. Si la composante en Z de chacune de ces normales sont relativement semblables, ceci indique que la surface est toujours orientée dans la même direction et qu'il y a donc peu de relief. Au contraire, plus les valeurs sont distinctes, plus la région est accidentée. Ainsi, pour chaque balayage, il suffit d'utiliser les normales pour en déterminer le niveau de relief. Deux techniques ont été testées

pour obtenir cette information.

## Variance des normales

Premièrement, la variance des normales donne directement une valeur représentative de la rugosité de la surface. Par exemple, avec les figures 4.6 et 4.7 représentent deux régions ayant un relief très différent. La variance des normales sont respectivement de 0.0335 et 0.0018. Cette différence notable permet de bien différencier les balayages entre eux et démontre que la variance des normales donne une bonne indication du niveau de relief.

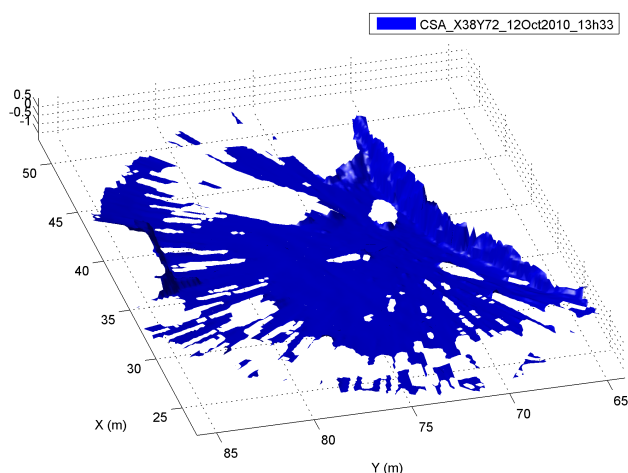


Figure 4.6 Exemple de balayage dans une région avec du relief

## Valeur seuil

Une autre méthode de calcul permet d'obtenir une valeur indiquant le niveau de relief à partir des normales. Il suffit de calculer le ratio de normales ayant une composante en Z plus grande qu'une certaine valeur seuil. Cette valeur de seuil a été déterminée empiriquement en comparant deux balayages les plus distincts possibles comme par exemple ceux des figures 4.6 et 4.7. En regardant l'histogramme des valeurs des normales représenté à la figure 4.8, nous remarquons que le balayage accidenté a des valeurs supérieures pour des normales inférieures à 0.99. C'est donc cette valeur qui est retenue comme seuil. Ainsi, l'indice du niveau de relief revient à calculer le rapport du nombre de normales supérieures à 0.99 sur le nombre total de points. À titre d'exemple, en reprenant les deux mêmes balayages, les valeurs obtenues sont de 67.35% et 90.21%.



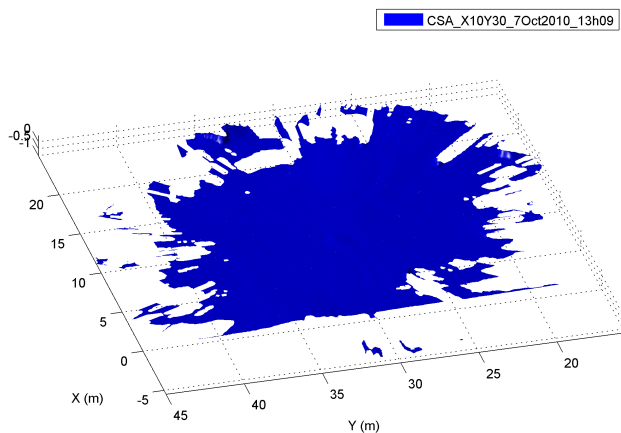


Figure 4.7 Exemple de balayage dans une région plane

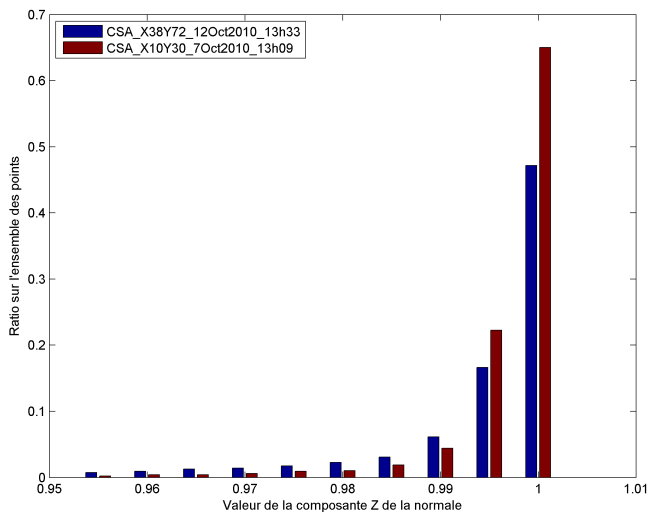


Figure 4.8 Histogramme représentant la composante Z des normales

#### 4.3.4 Distance de Hausdorff

Cette mesure a été présentée par Felix Hausdorff en 1914 pour calculer la distance entre deux sous-ensembles. Il s'agit de la plus grande distance entre un point et son plus proche voisin. Une manière plus intuitive de comprendre cette valeur est de s'imaginer un homme qui promène son chien en laisse. Les deux se déplacent chacun sur une surface. La distance d'Hausdorff correspond alors à la longueur minimale de la laisse. La figure 4.9 représente

aussi très bien cette valeur.

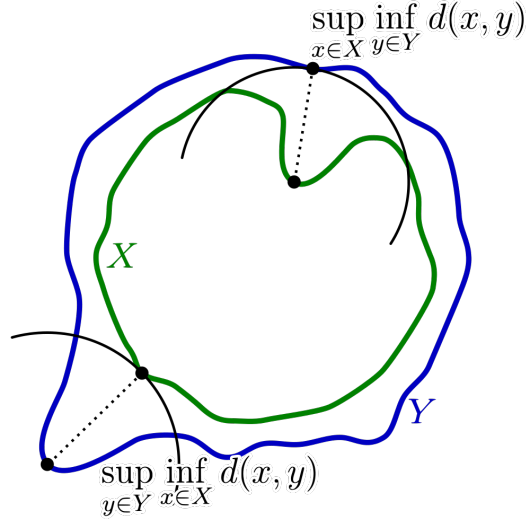


Figure 4.9 Représentation de la distance d’Hausdorff (Rocchini, 2007)

Cette distance n’a de sens que si elle est calculée sur la région où les deux balayages se superposent. Par contre, cette région est déterminée en rejetant les points dont leur plus proche voisin est plus loin d’un certain seuil, ce qui annule l’utilité de la distance d’Hausdorff puisque la plus grande distance ne sera jamais supérieure à ce seuil. Pour contrer ceci, nous avons fait la supposition que la surface est principalement dans le plan XY. Ainsi, la région de superposition est calculée en mesurant seulement les distances dans ce plan. Par la suite, la distance d’Hausdorff est calculée en utilisant les coordonnées des points dans les trois dimensions. Reprenons les balayages des figures 4.4 et 4.5 comme exemple. La région de superposition est sélectionnée avec une valeur seuil de 0.25 mètre, dans le plan XY. Par la suite, la distance d’Hausdorff pour chacun des cas est de 0.302 m et 0.566 m pour les positions réelles et erronées respectivement.

#### 4.3.5 « Spin image »

La technique de « spin image » est une technique bien connue dans le domaine de la reconnaissance d’objet. Elle consiste en un descripteur local, c’est-à-dire que c’est seulement la région au voisinage d’un point qui est comparée entre les balayages et non les balayages au complet. Pour appliquer la technique de « spin image », il suffit de choisir arbitrairement un point et de représenter tous les points dans son voisinage avec des coordonnées relatives à ce point central. Ainsi, ceux-ci n’auront plus les coordonnées spatiales XYZ, mais seront transformés dans un plan en deux dimensions dont les coordonnées sont la distance et l’élévation relatives au point central. Cette transformation est illustrée à la figure 4.10. Dans

cet exemple, le point central choisi est le point  $p$  et le point  $x$  étant dans son voisinage, ses nouvelles coordonnées  $\alpha$  et  $\beta$  sont calculées. Nous obtenons alors un nuage de points en deux dimensions représentant tous les voisins du point central choisi. Ensuite, nous effectuons une discrétisation de cet espace pour obtenir la « spin image » finale. La valeur de chaque pixel de l'image est calculée par interpolation bilinéaire comme illustré à la figure 4.11.

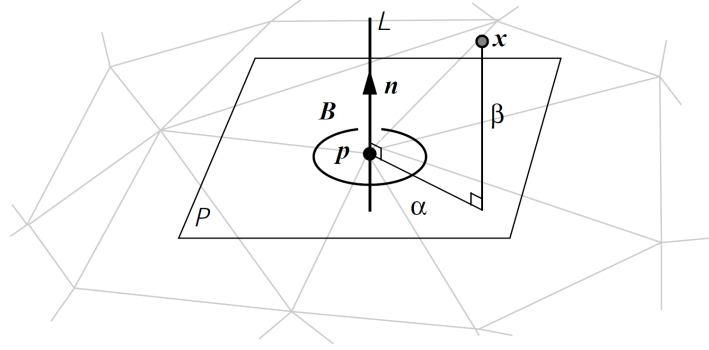


Figure 4.10 Transformation utilisée pour la technique « spin image » (Johnson, 1997)

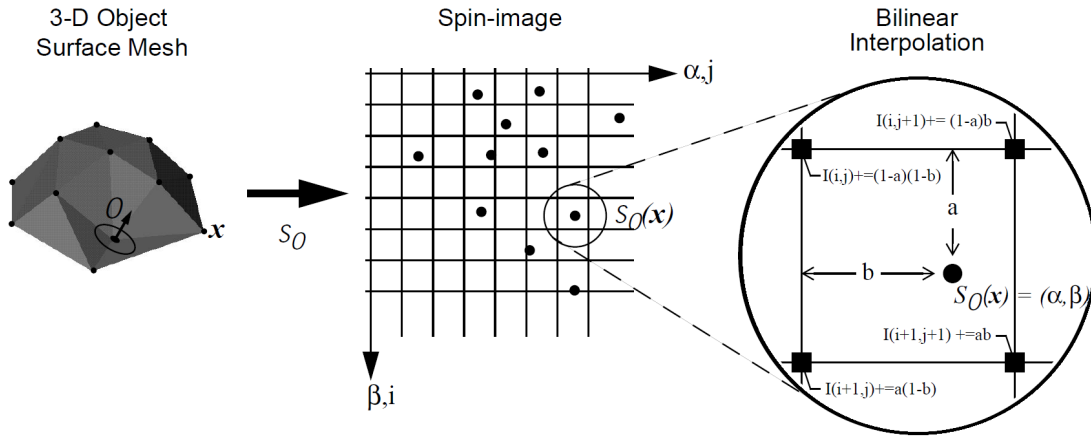


Figure 4.11 Étapes de la génération de « spin image » à partir d'un nuage de points (Johnson, 1997)

### Ajustement des paramètres

La qualité de l'image obtenue dépend fortement des valeurs utilisées pour la discrétisation. Si l'échantillonnage sur les axes  $\alpha$  et  $\beta$  sont trop fins, l'image résultante sera très sensible au bruit et deux images obtenues à partir de deux points rapprochés dans l'espace ne seront pas nécessairement semblables. Au contraire, si l'échantillonnage est trop large, nous

perdons beaucoup d'informations et l'image devient plus floue. Johnson (1997) propose un échantillonnage du même ordre de grandeur que la résolution du nuage de points. De plus, les valeurs maximales de  $\alpha$  et  $\beta$  ont un impact puisqu'ils déterminent la grandeur de la région au voisinage du point central qui sera transformée. Ainsi, nous avons quatre paramètres à déterminer qui auront un impact significatif dans la comparaison des « spin image ».

Pour déterminer ces paramètres, nous avons effectué des tests avec deux balayages ayant une bonne superposition. Ensuite, nous avons choisi au hasard cent points dans un des balayages et leur plus proche voisin dans l'autre. Nous avons ainsi cent paires de points dont les « spin image » peuvent être comparées. Ces images sont générées en testant différentes combinaisons de paramètres possibles.

Alternativement, nous avons sélectionné un autre point éloigné des deux autres et comparé l'image obtenue. L'objectif est de trouver les paramètres donnant les images les plus semblables pour les deux points rapprochés tout en donnant des images différentes de celle obtenue avec le point plus éloigné.

### Comparaison de « spin image »

La comparaison des images se fait par un calcul de coefficient de corrélation linéaire. Chaque pixel d'une image est comparée avec la valeur du pixel correspondant dans l'autre image. Ceci permet d'obtenir un diagramme de dispersion illustrant la ressemblance de deux images. Le coefficient de corrélation linéaire de ce diagramme permet de chiffrer cette ressemblance. L'équation 4.3 permet de calculer ce coefficient à partir de deux images  $P$  et  $Q$  contenant chacune  $i$  pixels.

$$R(P, Q) = \frac{N \sum p_i q_i - \sum p_i \sum q_i}{\sqrt{(N \sum p_i^2 - (\sum p_i)^2)(N \sum q_i^2 - (\sum q_i)^2)}} \quad (4.3)$$

Un exemple permettant d'illustrer la comparaison est présenté aux figures 4.12, 4.13 et 4.14. La première représente les deux nuages de points à comparer. Un point et son plus proche voisin dans l'autre balayage sont représentés par deux croix jaunes superposées. De plus, un troisième point, éloigné des autres, servira d'exemple pour une mauvaise comparaison.

Les deux autres figures illustrent les « spin images » et le diagramme de dispersion pour la bonne et la mauvaise correspondance. Sur la figure 4.13, les images sont plus semblables que sur la figure 4.14. Ceci est reflété par les coefficients obtenus qui sont de 0.8992 et 0.5975 pour la bonne et la mauvaise correspondance respectivement.

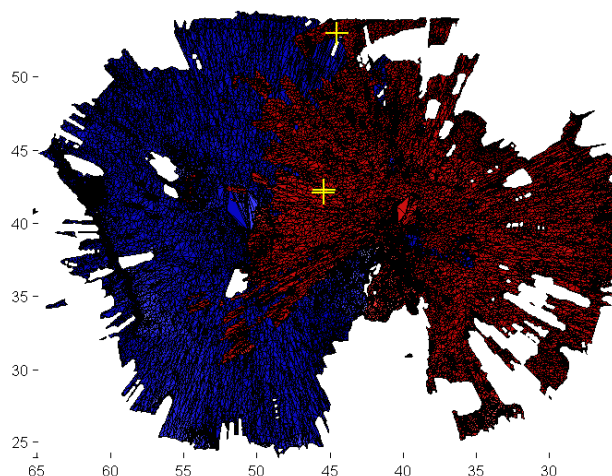


Figure 4.12 Exemple de points utilisés pour comparer les images

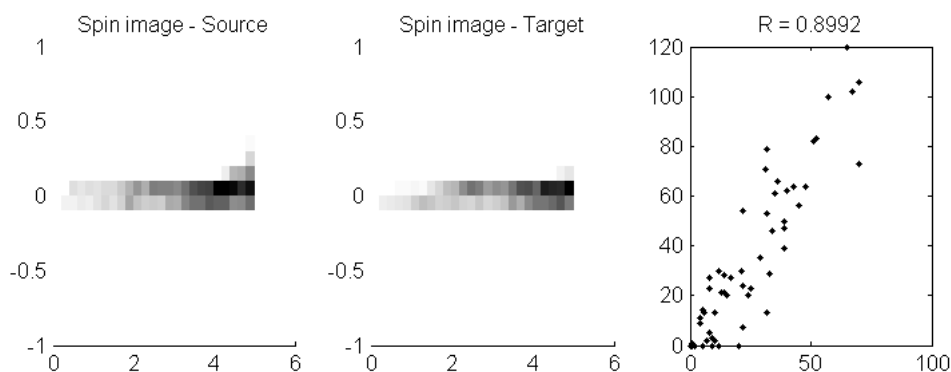


Figure 4.13 Images produites et diagramme de dispersion pour une bonne correspondance

#### 4.3.6 Image de distances (« Range image »)

La technique de « spin image » donne un descripteur local autour d'un point choisi au hasard. Il serait intéressant d'avoir une technique semblable mais permettant d'obtenir un descripteur global. C'est l'utilité d'une image de distance. La figure 4.16 représente l'image de distance du balayage représenté à la figure 4.15. Chacun des pixels de cette image représente une certaine région du balayage dans le plan XY. La moyenne de la composante Z de ces points donne une valeur au pixel.

La même technique de comparaison utilisée pour les « spin images » est applicable dans

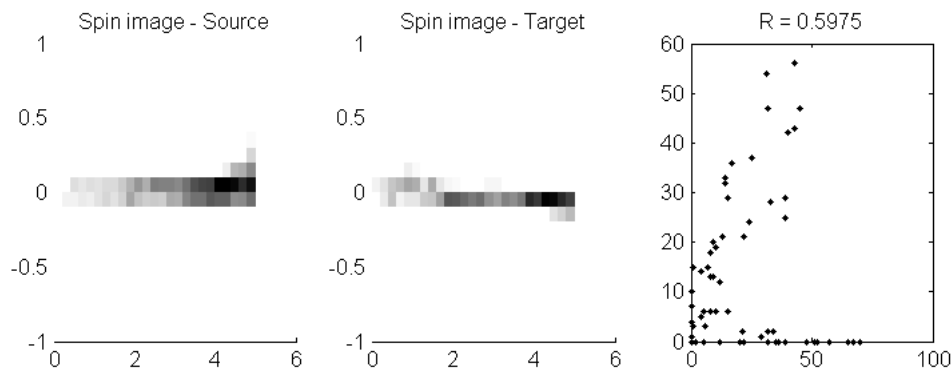


Figure 4.14 Images produites et diagramme de dispersion pour une mauvaise correspondance

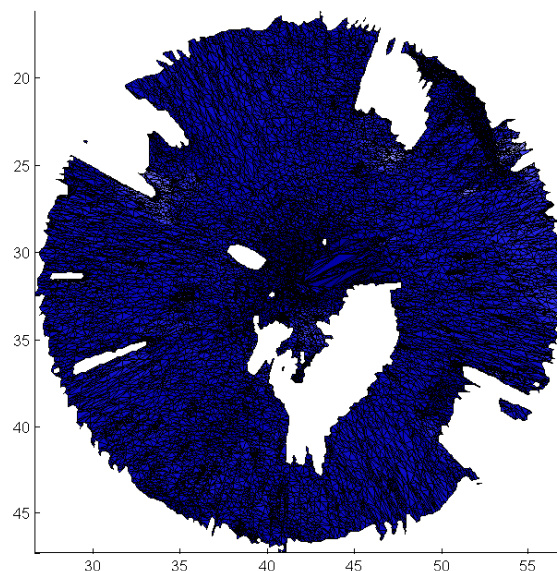


Figure 4.15 Balayage utilisé pour l'exemple d'image de distance

le cas des images de distance. Il suffit d'éliminer les points n'étant pas dans la surface de superposition et de générer une image de distance pour chacun des balayages. Finalement, le coefficient de corrélation linéaire du diagramme de dispersion donne une valeur indiquant la ressemblance entre les deux balayages. Plus ceux-ci sont semblables, plus ils ont de chances de représenter la même surface.

De plus, l'index MSSIM (Mean Structural Similarity) est calculé comme dans les travaux de Wang *et al.* (2004). Cette valeur consiste en la comparaison de la luminance, du contraste et de la structure des deux images.

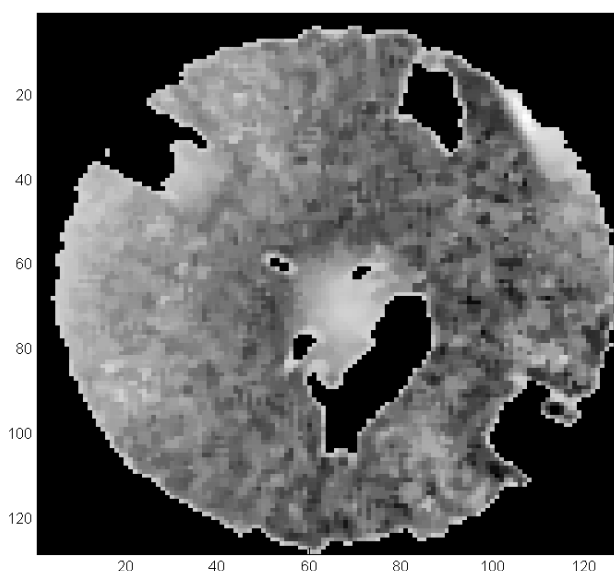


Figure 4.16 Exemple d'image de distance

#### 4.3.7 Distance de chaque point avec son plus proche voisin

Finalement, une dernière valeur qui a été testée correspond à la distance de chaque point avec son plus proche voisin dans l'autre balayage. L'hypothèse est que pour un bon recalage, les surfaces se superposent presque parfaitement et la majorité des points de la région de superposition sont très près du point de l'autre surface. Par contre, si le résultat du recalage est erroné, certains points devraient être plus éloignés de l'autre surface. Ainsi, en déterminant une valeur seuil, il est possible d'obtenir le pourcentage des points dans la région de superposition qui ont un plus proche voisin en deçà de cette valeur seuil. Plusieurs tests ont été fait avec des valeurs variant de 0.1 à 1 mètre. Quand cette valeur est aux alentours de 0.4 mètre, les résultats obtenus pour un bon et un mauvais recalage sont le plus différentiable. Les figures 4.17 et 4.18 représentent respectivement un bon et un mauvais résultat de recalage obtenus dans la même région. Pour le bon résultat près de 94.9% des points de la région de superposition ont un plus proche voisin à moins de 0.4 mètre. Ce pourcentage diminue à 86.9% dans l'autre cas. Ceci indique que les surfaces se superposent mieux dans le premier cas.

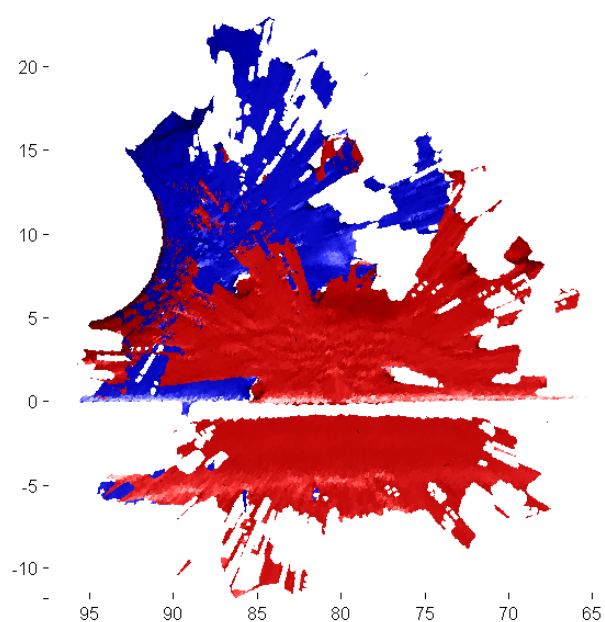


Figure 4.17 Bon recalage, pourcentage de points ayant un PPV à moins de 0.4m : 94.9%

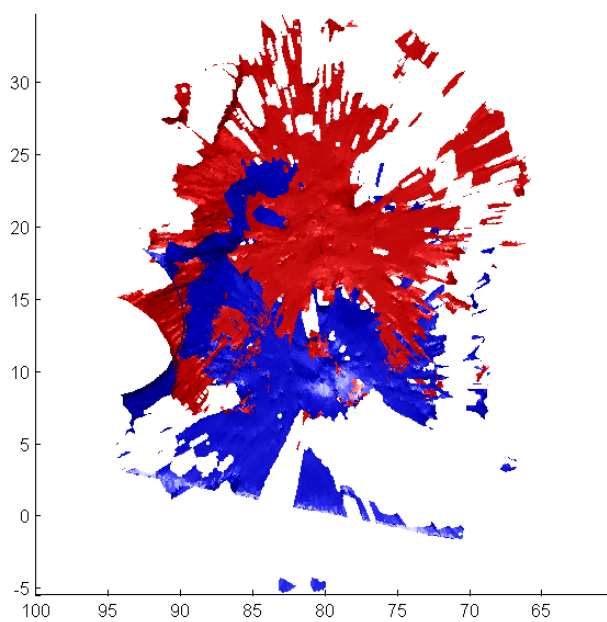


Figure 4.18 Mauvais recalage, pourcentage de points ayant un PPV à moins de 0.4m : 86.9%



## CHAPITRE 5

### DONNÉES EXPÉRIMENTALES

Toutes les données expérimentales utilisées dans ce projet ont été acquises sur le terrain d'émulation de Mars, « Mars Emulation Terrain » (MET), situé à l'Agence Spatiale Canadienne. Les mesures se font à l'aide d'un LIDAR installé soit sur un robot mobile ou tout simplement une plate-forme mobile. Un LIDAR (« **L**ight **d**etection and **r**anging ») fonctionne selon le même principe qu'un radar, mais avec un Laser plutôt qu'une onde radio. Ainsi, un faisceau laser est émis et lorsque celui-ci frappe un objet, il est réfléti et retourne vers le LIDAR. En mesurant l'intensité du faisceau reçu, il est possible de calculer la distance de l'objet rencontré. Finalement, les angles d'azimut et d'élévation utilisés pour envoyer le faisceau laser complètent les coordonnées sphériques du point en question. Une étape de post-traitement est requise pour filtrer les données, les convertir en coordonnées cartésiennes et obtenir la triangulation. Trois phases d'essais ont été faites au cours desquelles différents montages ont été utilisés.

#### 5.1 Terrain d'émulation de Mars

Le terrain d'émulation de Mars de l'ASC, figure 5.1, a été spécialement conçu pour le développement de robots mobiles. Ce terrain de 120 mètres par 60 mètres est principalement composé de sable, roches de différents formats, cratères, etc. Deux régions sont particulièrement intéressantes pour le problème de recalage, soit la « saddle mountain » et la région autour de la plaque de dalles Alpha (« flag stone patche Alpha »).

##### 5.1.1 « Saddle Mountain »

Cette colline est constituée de deux sommets et d'un passage au milieu pour le robot. Cette partie du terrain est intéressante puisqu'elle permet de simuler une trajectoire en boucle fermée autour de cette colline. Les parois de la collines ont l'avantage de fournir un relief assez irrégulier facilitant le recalage. Par contre, elles bloquent la vue du robot et limitent ainsi la superposition de chaque balayage avec les autres. Une autre caractéristique intéressante est le passage au milieu de la colline permettant de faire une trajectoire en huit et ainsi de créer plus de dépendances entre les recalages pour l'algorithme de boucle fermée. La figure 5.2 montre une vue aérienne de cette région du terrain et la figure 5.3 donne un exemple de balayage incluant une partie de ce mont.



Figure 5.1 Terrain d'émulation de Mars (CSA, 2011)



Figure 5.2 « Saddle Mountain » (CSA, 2011)

### 5.1.2 Plaque de dalles Alpha

Cette partie du terrain d'émulation de Mars ne contient pratiquement aucune aspérité. Ceci implique qu'il est très improbable d'obtenir un bon recalage si les conditions initiales

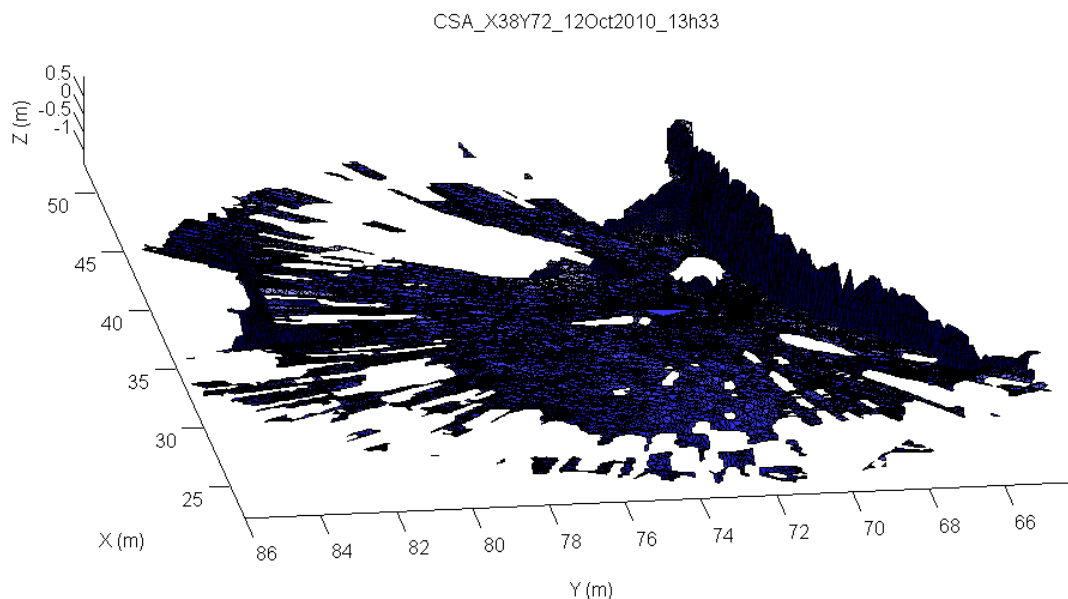


Figure 5.3 Exemple de balayage près de « Saddle Mountain » (CSA, 2011)

sont le moins erronées. C'est donc une région critique où les résultats donnés par l'algorithme ICP sont très incertains. C'est un exemple typique de moment où une bonne métrique d'erreur de recalage est nécessaire pour donner une probabilité de bonne convergence et utiliser ces informations dans l'algorithme de boucle fermée. La figure 5.4 représente deux balayages effectués dans cette région. On imagine très bien que les surfaces sont très difficiles à recalcr, même avec une intervention humaine.

## 5.2 Données de « Mobile Robotics Test-Bed »

Les données utilisées pour ce projet sont celles qui ont été acquises par le « Mobile Robotics Test-Bed (MRT) » de l'ASC qui composent leur base de données appelée PERLdb (« Planetary Exploration Rover LIDAR Database »). Cette base de données contient 102 balayages acquis avec différentes résolutions. La plus petite résolution, celle qui a été utilisée, est de 0.33 degrés d'azimut. L'acquisition des données a été faite du 29 septembre 2010 au 13 octobre 2010 et le montage expérimental consistait en un LIDAR (SICK LMS291), trois antennes GPS et une unité de mesure d'inertie (IMU). Cette dernière fournit l'orientation du robot tandis que les trois antennes GPS en plus d'une quatrième fixée sur le toit du hangar forment un GPS différentiel (RTK DGPS). Ainsi, pour chaque balayage effectué par le LIDAR, la position et l'orientation du robot sont connues avec une grande précision.

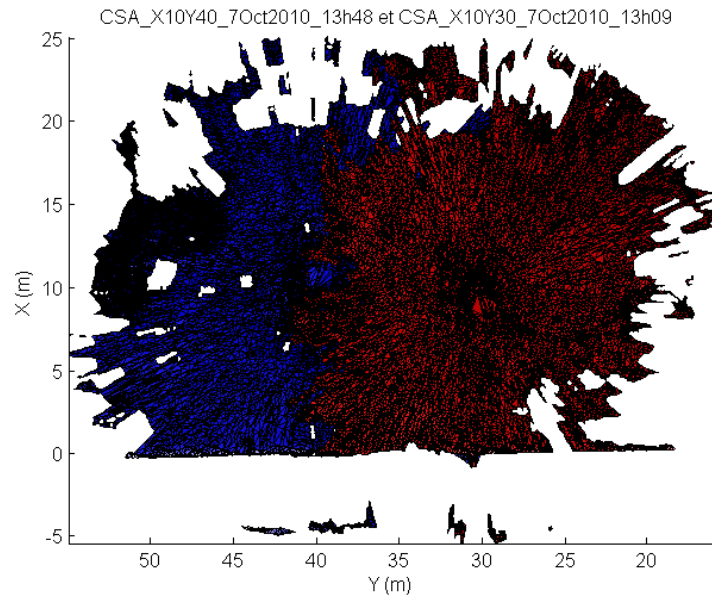


Figure 5.4 Exemple de deux balayages près de la plaque de dalles Alpha (CSA, 2011)

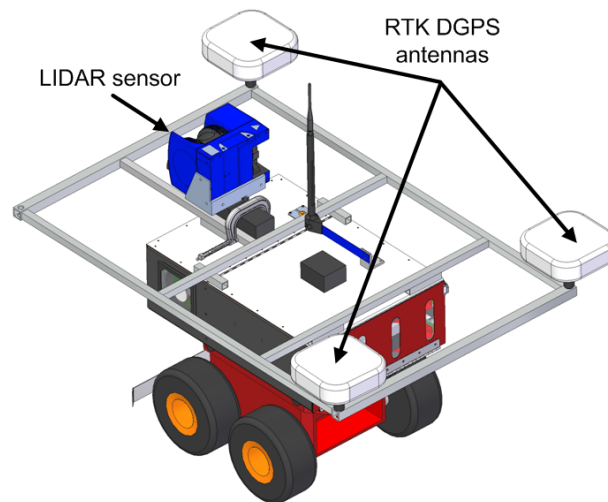


Figure 5.5 Montage expérimental du « Mobile Robotics Test-Bed » (CSA, 2011)

## CHAPITRE 6

### RÉSULTATS

#### 6.1 Métrique d'erreur

À partir des techniques présentées dans le dernier chapitre, une liste de 22 variables a été définie pour utiliser dans l'algorithme de discrimination (Section 4.2). Ces variables sont présentées au tableau 6.1<sup>1</sup>. Chacune des combinaisons possibles a été testée dans l'algorithme de discrimination et celles ayant obtenu le meilleur APER ont été retenues. Ceci donne un total de 2097152 ( $2^{21}$ ) cas comparés pour en déterminer la solution optimale.

La figure 6.1 représente les 10000 combinaisons donnant les meilleurs APER. Le cas optimal est celui composé des paramètres suivants :

- Le pourcentage de superposition selon la méthode de Chetverikov *et al.* (2005)
- Le nombre d'itérations
- La distance d'Hausdorff
- Les normales des balayages source et cible
- Le coefficient R des « spin image »
- Le nombre de points finis dans les images de distance

Avec ces paramètres, le taux apparent d'erreur est de 2.12%, ce qui correspond à 20 faux négatifs et 69 faux positifs sur les 4203 recalages utilisés. Cependant, il est intéressant de regarder les autres combinaisons donnant de bons APER pour voir l'impact de chacun des paramètres et ainsi noter ceux qui ont plus d'importance que d'autres. La figure 6.2 donne le nombre de fois où chaque variable est utilisée dans les 10000 cas donnant le meilleur APER.

Il est intéressant de noter qu'aucune variable n'est utilisée dans 100 % des cas. Par contre, quelques-unes sont utilisées dans plus de 2/3 des cas. Il s'agit du pourcentage de superposition, la distance d'Hausdorff, le coefficient R des « spin image » ainsi que certaines normales. À l'inverse, le pourcentage de plus proches voisins inférieurs à 0.4 mètres ainsi que la variance des distances des plus proches voisins sont utilisés très rarement et apportent donc peu d'information.

---

1. Normale 1 correspond au relief des balayage calculé par la variance des normales et Normale 2 correspond au relief calculé par la valeur seuil (section 4.3.3).

Tableau 6.1 Liste des variables utilisées dans l'algorithme de discrimination

No	Variable
1	Pourcentage de superposition calculé par l'algorithme
2	Pourcentage de superposition selon la méthode de Chetverikov <i>et al.</i> (2005)
3	Nombre d'itérations
4	Distance d'Hausdorff
5	Normale 1 de tout le balayage source
6	Normale 1 de tout le balayage cible
7	Normale 2 de tout le balayage source
8	Normale 2 de tout le balayage cible
9	Normale 1 de la région de superposition du balayage source
10	Normale 1 de la région de superposition du balayage cible
11	Normale 2 de la région de superposition du balayage source
12	Normale 2 de la région de superposition du balayage cible
13	Coefficient R de « Spin Image »
14	Coefficient C de « Spin Image »
15	Nombre de points finis des images de distance
16	Erreur quadratique moyenne des images de distance
17	Coefficient R des images de distance
18	MSSIM des images de distance
19	Pourcentage de plus proches voisins du balayage source inférieurs à 0.4 mètre
20	Pourcentage de plus proches voisins du balayage cible inférieurs à 0.4 mètre
21	Variance de la distance des points du balayage source avec leur plus proche voisin
22	Variance de la distance des points du balayage cible avec leur plus proche voisin

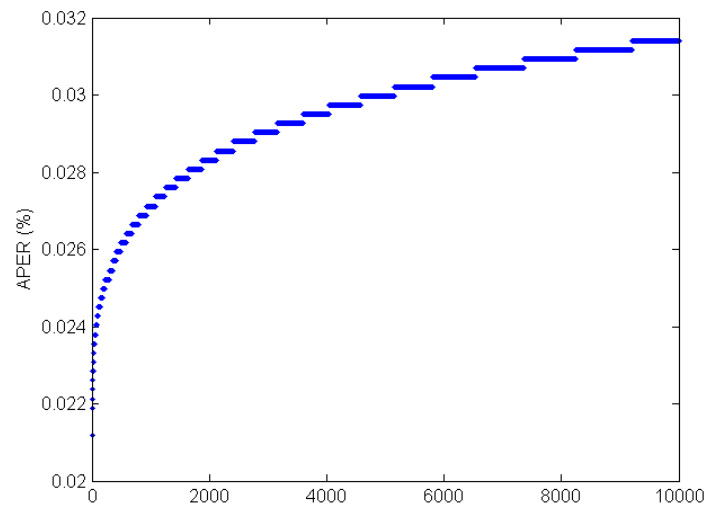


Figure 6.1 APER des 10000 meilleures combinaisons de paramètres

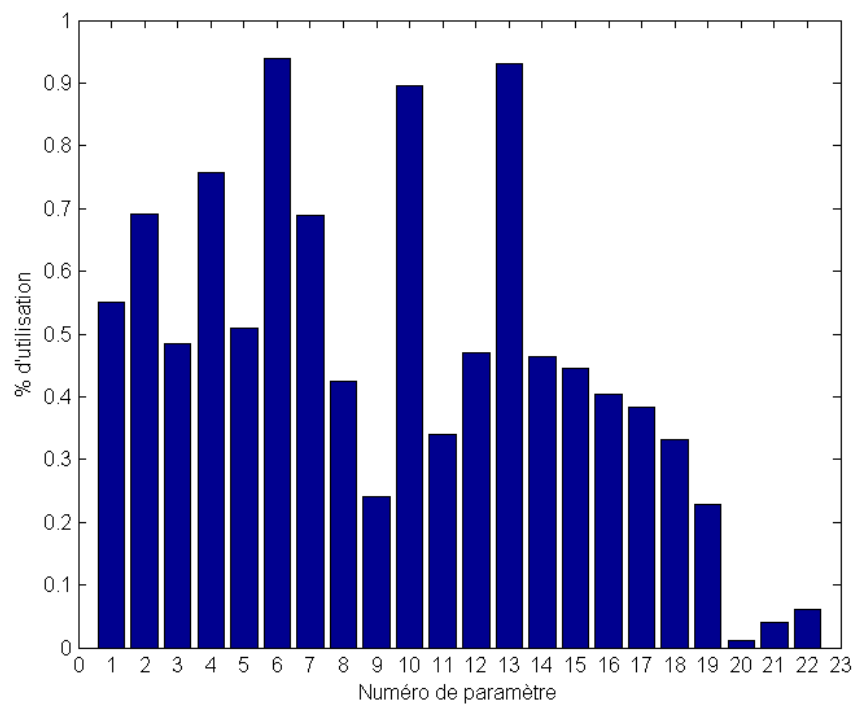


Figure 6.2 Pourcentage d'utilisation de chaque variable dans les 10000 meilleures combinaisons

## 6.2 Étude du temps de calcul

Quoique la combinaison donnant le meilleur taux apparent d'erreur a été déterminée, il est intéressant de regarder la complexité de calcul de chaque paramètre pour ainsi déterminer si une autre combinaison pourrait donner un APER presque aussi bon, mais étant composé de variables plus simples à calculer. De plus, le nombre de paramètres a une importance dans le calcul même de la métrique où une fonction de distribution gaussienne multivariable est utilisée 4.1.

Certains paramètres n'ont besoin d'aucun calcul supplémentaire puisqu'ils sont déjà disponibles à partir de l'algorithme ICP. C'est le cas du pourcentage de superposition, du nombre d'itérations et des normales de chaque point. Nous pouvons donc utiliser ces variables dans la métrique pratiquement sans ajouter de temps de calcul. Le calcul des plus proches voisins est lui aussi fait dans l'algorithme de recalage. Cependant, il est effectué sur un sous-ensemble des points contrairement à tous les points comme fait pour les calculs des paramètres. Il faut donc refaire ces calculs pour en extraire notamment la région de superposition qui est importante pour plusieurs paramètres. Le tableau 6.2 liste les temps de calcul pour chaque variable à l'exception de celles fournies directement par l'algorithme de recalage. Par contre, il faut noter que ces temps de calcul n'ont pas été optimisés.

Tableau 6.2 Temps de calcul moyen pour chaque variable

Variable	Temps (s)
Pourcentage de superposition selon la méthode de Chetverikov <i>et al.</i> (2005)	1.1588
Distance d'Hausdorff	0.0024
Normale 1 de tout le balayage	0.0069
Normale 1 de la région de superposition des balayages	0.0009
Normale 2 de tout le balayage	0.0003
Normale 2 de la région de superposition des balayages	0.0006
« Spin Image »	5.7811
Images de distance	3.4083
Pourcentage de plus proches voisins inférieurs à 0.4 mètre	0.0017
Variance de la distance des points avec leur plus proche voisin	0.0031

Le pourcentage de superposition, les « spin images » et les images de distance sont les variables demandant le plus de temps de calcul. Concernant le pourcentage de superposition, le temps est d'un peu plus de 1 seconde puisqu'il faut recalculer la distance de chaque point d'un balayage avec son plus proche voisin. Quant aux « spin images » et aux images de distance, leurs temps de calcul supérieurs s'expliquent par des algorithmes plus complexes

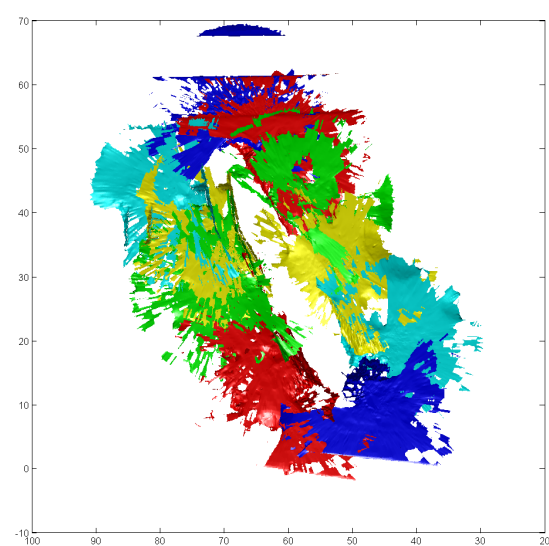


qui n'ont pas été optimisés.

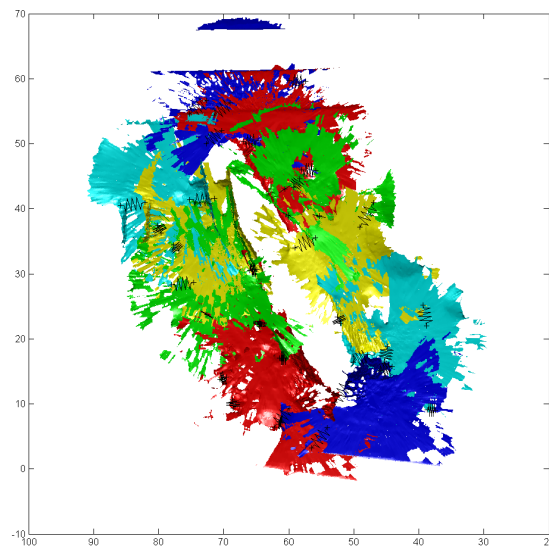
La meilleure combinaison n'utilisant aucun paramètre de 13 à 18 donne un APER de 7.23% avec 16 faux négatifs et 288 faux positifs. Notons que les faux positifs sont plus graves que les faux négatifs puisqu'il est pire de considérer un mauvais recalage que d'exclure un bon recalage. Nous en concluons que les « spin images » et les images de distance sont très importants dans la métrique développée. Il faut donc faire le compromis entre les temps de calcul et la précision de la métrique. Il est important aussi de noter qu'une optimisation des algorithmes plus complexes permettrait de réduire leur désavantage relié au temps de calcul.

### 6.3 Simulation de l'algorithme de boucle fermée

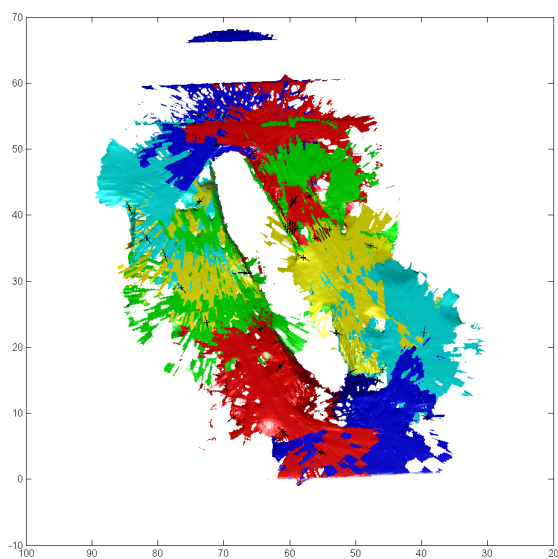
Le but d'établir une métrique d'erreur de recalage est de pouvoir utiliser celle-ci dans l'algorithme de boucle fermée présenté à la section 3.2. La figure 6.3 représente un exemple de cet algorithme utilisé dans un cas où le robot effectue une boucle fermée mais un recalage intermédiaire est erroné. Le balayage vert affiché au coin supérieur droit de la trajectoire a une erreur de rotation trop grande qui a mené à un mauvais recalage avec le balayage en jaune. Si la métrique donne une faible valeur pour ce recalage, l'algorithme de répartition de l'erreur permet alors d'accorder moins d'importance à cette information et utiliser les bons recalages pour créer une carte globale. Sur la figure 6.3(c), nous pouvons voir que les ressorts reliant les deux balayages en question sont plus détendus que les autres, permettant d'obtenir un meilleur résultat. En comparant cette image avec 6.3(d) représentant la carte réelle, nous pouvons constater que le résultat est très près de la réalité. En fait, l'erreur moyenne pour l'ensemble des balayages est de 0.41 mètre et 0.58 radians comparativement à 0.19 mètres et 0.51 radians pour le cas idéal présenté au chapitre 3.



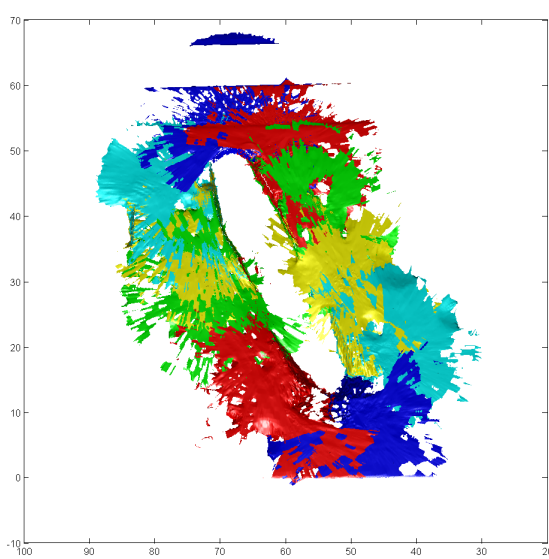
(a) Conditions initiales



(b) Ressorts fixés



(c) Itération 2



(d) Itération 5

Figure 6.3 Exemple de recalage

## CHAPITRE 7

### CONCLUSION

#### 7.1 Synthèse des travaux

Ce projet est composé de deux parties principales. Premièrement, un algorithme de distribution d'erreur a été programmé et testé avec des données fournies par l'Agence Spatiale Canadienne représentant le plus fidèlement possible le type de surface rencontré sur Mars. Cette méthode est basée sur les travaux de Eggert *et al.* (1996) et consiste à apposer des ressorts virtuels entre chaque paire de balayage. Ces ressorts composent ainsi un système dynamique convergeant naturellement vers la situation représentant le minimum d'énergie. Ceci permet de redistribuer l'erreur accumulée sur l'ensemble d'une trajectoire et ainsi d'obtenir une carte globale pour uniforme. Cependant, nous avons rapidement conclu que cet algorithme est très sensible au résultat de chaque recalage. Une erreur à cette étape est très difficilement détectée et diminue de beaucoup la précision de la carte globale acquise.

Ces constatations justifient la deuxième partie du projet qui consiste à déterminer une métrique d'erreur permettant de quantifier la probabilité de la validité du résultat d'un recalage. Plusieurs paramètres ont été testés et aucun d'entre eux ne permet à lui seul de séparer les bons des mauvais résultats. Les paramètres testés sont expliqués dans le chapitre 4 et sont rassemblés dans le tableau 6.1. Nous avons donc conçu un algorithme de discrimination qui, à partir d'une base de données de plusieurs résultats de recalages, modélise une fonction de distribution gaussienne multivariable pour chacun des deux groupes (bons et mauvais recalages). Par la suite, le résultat d'un nouveau recalage est comparé à chacun des groupes et est classé parmi celui y ressemblant le plus. La meilleure combinaison de paramètres a permis d'obtenir un taux d'erreur de classification de 2.12% avec 20 faux négatifs et 69 faux positifs.

#### 7.2 Limitations de la solution proposée et améliorations futures

Certaines approximations ont été faites lors de ce projet. Notamment, la détection d'une trajectoire fermée est un tout autre problème en soi mais qu'il est nécessaire de résoudre pour l'application de l'algorithme de répartition d'erreur. Cette détection a été considérée comme étant déjà faite au préalable. Une autre limitation est le fait que tout le projet a été effectué avec les données acquises par un seul type de LIDAR. Les balayages ont alors tous les mêmes caractéristiques et il serait intéressant de tester ces techniques avec d'autres types de données. Notamment, les angles totaux des balayages n'étaient pas assez grands pour pouvoir avoir

des trajectoires avec beaucoup de redondances, c'est-à-dire plusieurs balayages se superposant. Ceci apporterait des contraintes supplémentaires dans l'algorithme de distribution et permettrait forcément d'être plus robuste vis-à-vis des mauvais recalages.

Aussi, comme mentionné dans le chapitre 6, le calcul des plus proches voisins se fait avec l'ensemble des points, ce qui permet d'obtenir plus d'information qu'avec un sous-échantillon, mais demande plus de temps de calcul. Des tests concernant le calcul des paramètres de la métrique avec seulement un sous-échantillon pourraient permettre de réduire la charge de calcul sans trop réduire la précision de la métrique. De plus, les calculs de certains paramètres comme les « spin images » et les images de distance n'ont pas été optimisés.

Finalement, la métrique n'a pas été testée sur le terrain. L'objectif final étant d'intégrer ces méthodes pour améliorer l'autonomie des robots mobiles, des tests en situation réelle permettraient de mettre au point les techniques et d'en détecter des limitations. Néanmoins, la métrique proposée permettant de classifier les résultats des recalages avec un taux d'erreur d'un peu plus de 2% apporte un outil supplémentaire pour l'augmentation de l'autonomie des robots mobiles en environnement inconnu.

## RÉFÉRENCES

- BARNEA, S. et FILIN, S. (2008). Keypoint based autonomous registration of terrestrial laser point-clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63, 19 – 35. Theme Issue : Terrestrial Laser Scanning.
- BESL, P. et MCKAY, H. (1992). A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14, 239 –256.
- CHEN, Y. et MEDIONI, G. (1991). Object modeling by registration of multiple range images. *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. vol. 3, 2724 –2729.
- CHETVERIKOV, D., STEPANOV, D. et KRSEK, P. (2005). Robust euclidean alignment of 3d point sets : the trimmed iterative closest point algorithm. Elsevier, vol. 23, 299–309.
- CHOE, T. E. et MEDIONI, G. (2007). 3-d metric reconstruction and registration of images of near-planar surfaces. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. 1 –8.
- COLE, D. et NEWMAN, P. (2006). Using laser range data for 3d slam in outdoor environments. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. 1556 –1563.
- CSA (2011). *Planetary Exploration Rover LIDAR Database*. Base de données de l’Agence Spatial Canadienne.
- DU, S., ZHENG, N., YING, S. et LIU, J. (2010). Affine iterative closest point algorithm for point set registration. *Pattern Recognition Letters*, 31, 791 – 799.
- EGGERT, D., FITZGIBBON, A. et FISHER, R. (1996). Simultaneous registration of multiple range views for use in reverse engineering. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. vol. 1, 243 –247.
- GEMME, S., GINGRAS, D., SALERNO, A., DUPUIS, E., POMERLEAU, F. et MICHAUD, F. (2012). Pose refinement using icp applied to 3-d lidar data for exploration rovers. *Proc. 2012 International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- JOHNSON, A. E. (1997). Spin-images : a representation for 3-d surface matching.
- LINGEMANN, K., NÜCHTER, A., HERTZBERG, J. et SURMANN, H. (2005). High-speed laser localization for mobile robots. *Robotics and Autonomous Systems*, 51, 275 – 296.

- LIU, Y. (2006). Automatic registration of overlapping 3d point clouds using closest points. *Image and Vision Computing*, 24, 762 – 781.
- LIU, Y. (2010). Automatic range image registration in the markov chain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32, 12 –29.
- NASA (2013). *Jet Propulsion Laboratory California Institute of Technology*. <http://www.jpl.nasa.gov/spaceimages/details.php?id=PIA16934>.
- NEUGEBAUER, P. (1997). Geometrical cloning of 3d objects via simultaneous registration of multiple range images. *Shape Modeling and Applications, 1997. Proceedings., 1997 International Conference on*. 130 –139.
- NÜCHTER, A., LINGEMANN, K., HERTZBERG, J. et SURMANN, H. (2007). 6d slam-3d mapping outdoor environments. *Journal of Field Robotics*, 24, 699 – 722.
- OHNO, K., NOMURA, T. et TADOKORO, S. (2006). Real-time robot trajectory estimation and 3d map construction using 3d camera. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. 5279 –5285.
- PULLI, K. (1999). Multiview registration for large data sets. *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*. 160 –168.
- ROCCHINI (2007). *Wikipedia : Hausdorff Distance*. [http://en.wikipedia.org/wiki/Hausdorff\\_distance](http://en.wikipedia.org/wiki/Hausdorff_distance).
- RUSINKIEWICZ, S. et LEVOY, M. (2001). Efficient variants of the icp algorithm. *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. 145 –152.
- SHARP, G., LEE, S. et WEHE, D. (2002). Icp registration using invariant features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24, 90 –102.
- SURMANN, H., NÜCHTER, A. et HERTZBERG, J. (2003). An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45, 181 – 198.
- VIEJO, D. et CAZORLA, M. (2007). 3d plane-based egomotion for slam on semi-structured environment. *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. 2761 –2766.
- WANG, F., VEMURI, B., RANGARAJAN, A. et EISENSCHENK, S. (2008). Simultaneous nonrigid registration of multiple point sets and atlas construction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30, 2011 –2022.
- WANG, Z., BOVIK, A. C., SHEIKH, H. R. et SIMONCELLI, E. P. (2004). Image quality assessment : From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13, 600–612.

## ANNEXE A

### Variables

Cet annexe rassemble les graphiques de chacune des variables utilisées dans l'algorithme de discrimination. Ces variables sont affichées en fonction de l'erreur en translation et en rotation du recalage. Les points bleus et les points rouges représentent les bons et les mauvais résultats respectivement. Ces graphiques démontrent notamment qu'aucune de ces variables permet à elle seule de faire une bonne classification.

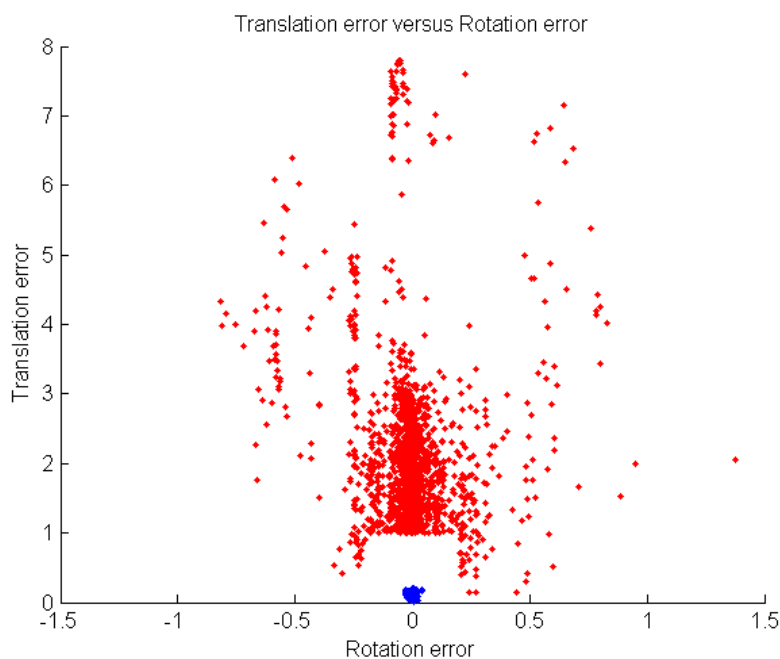


Figure A.1 Erreurs de translation en fonction de l'erreur de rotation

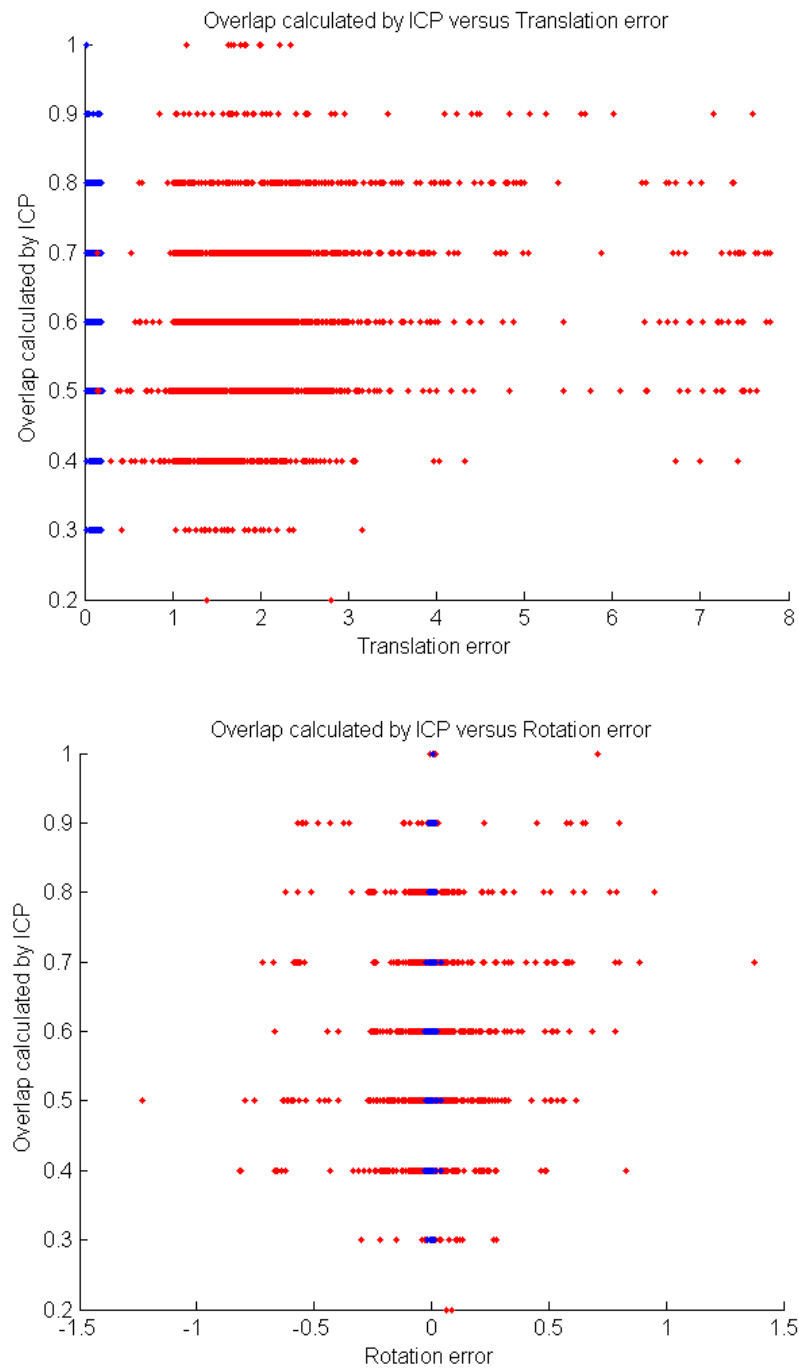


Figure A.2 Région de superposition calculée par l'algorithme ICP



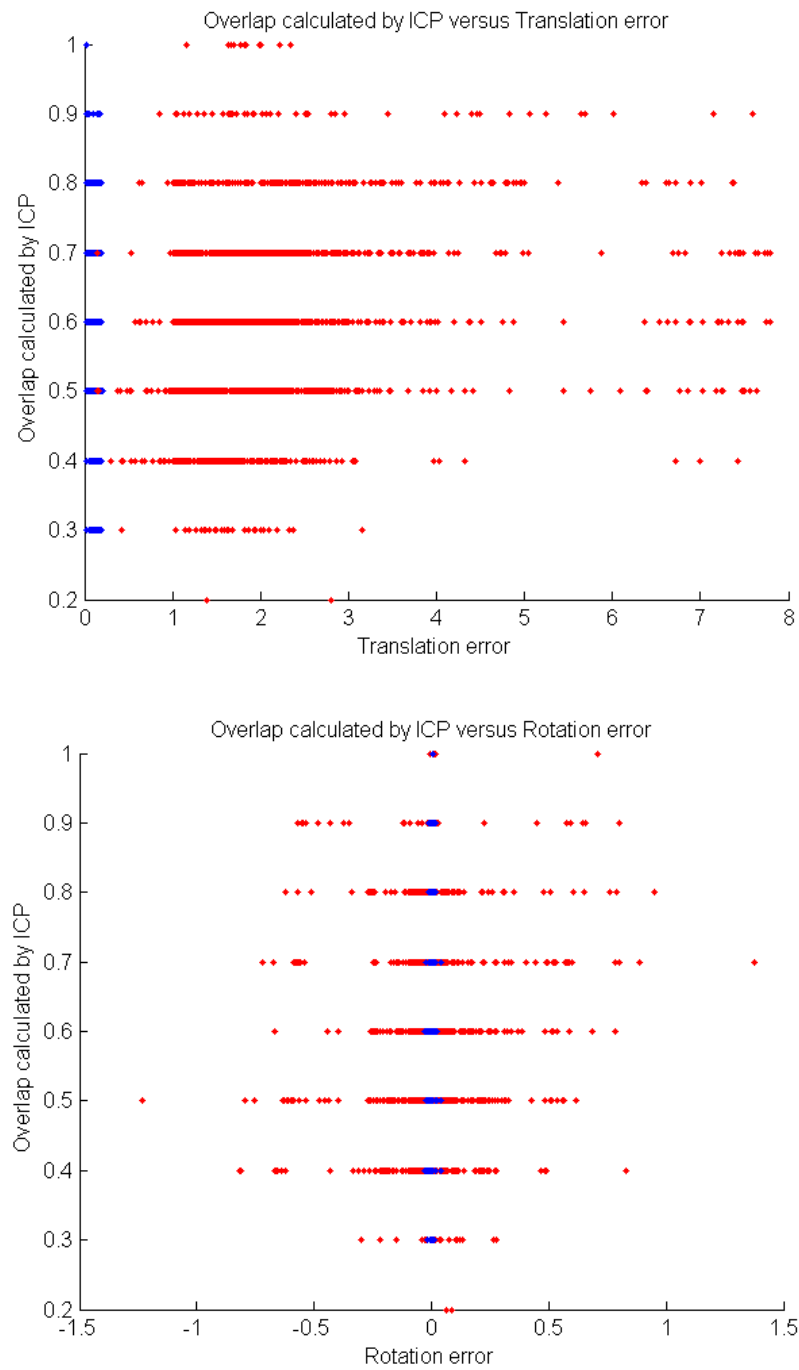


Figure A.3 Région de superposition calculée par l'algorithme ICP

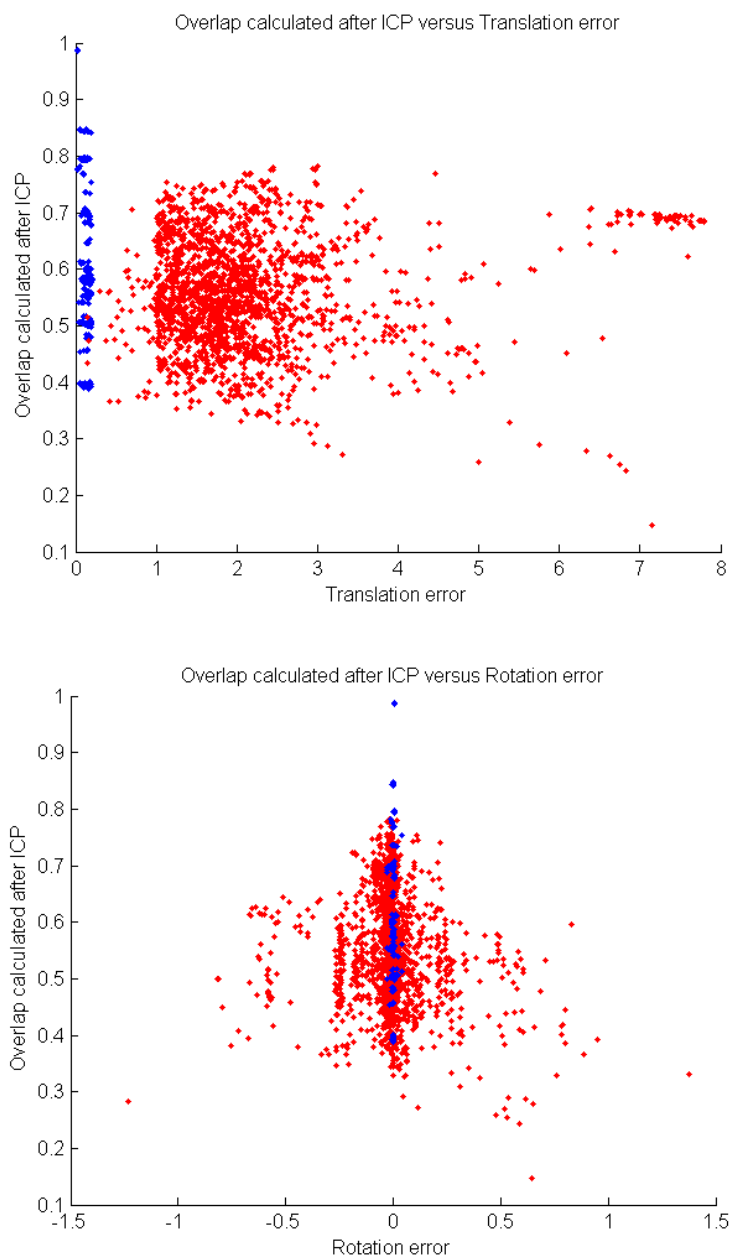


Figure A.4 Région de superposition calculée selon la méthode de Chetverikov *et al.* (2005)

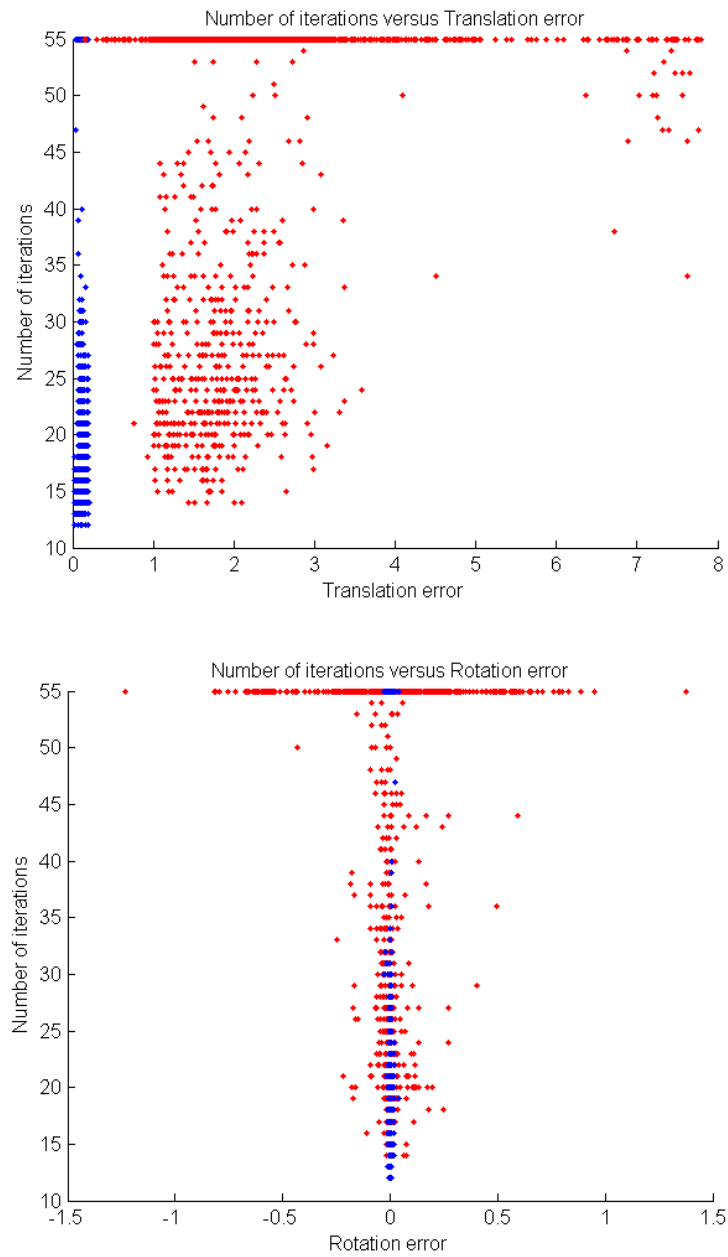


Figure A.5 Nombre d'itérations

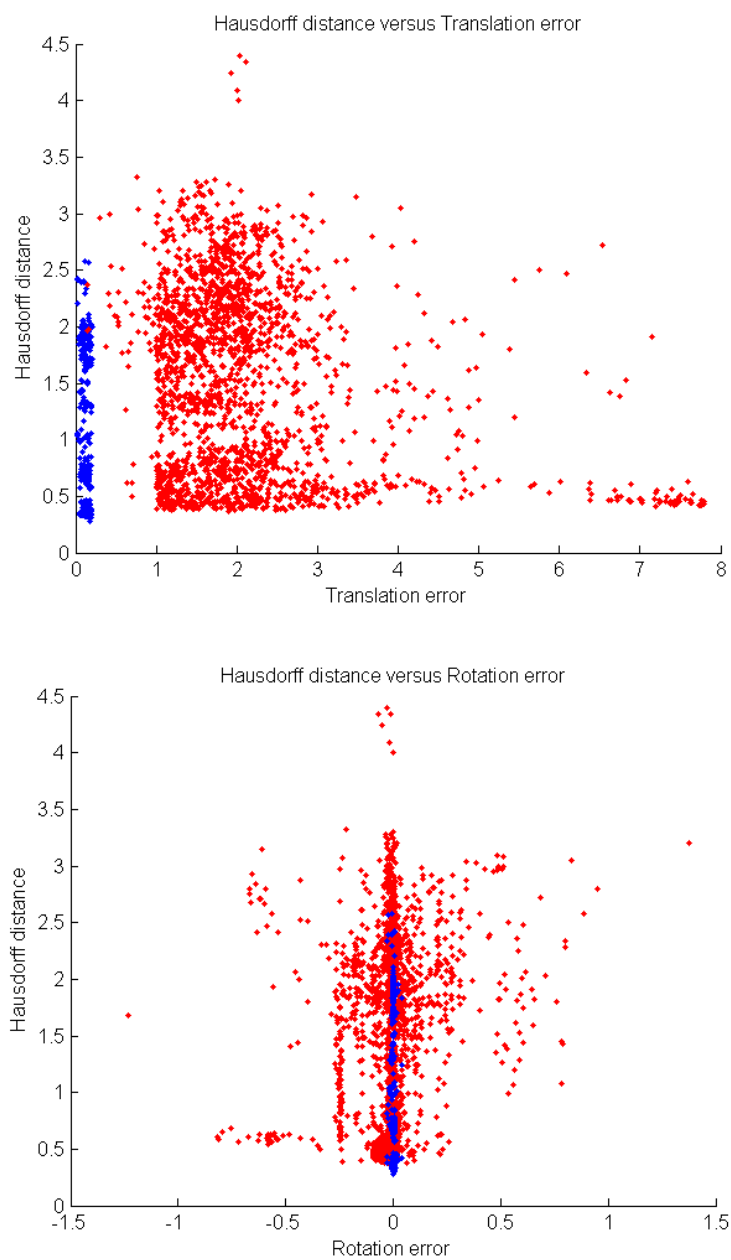


Figure A.6 Distance d'Hausdorff

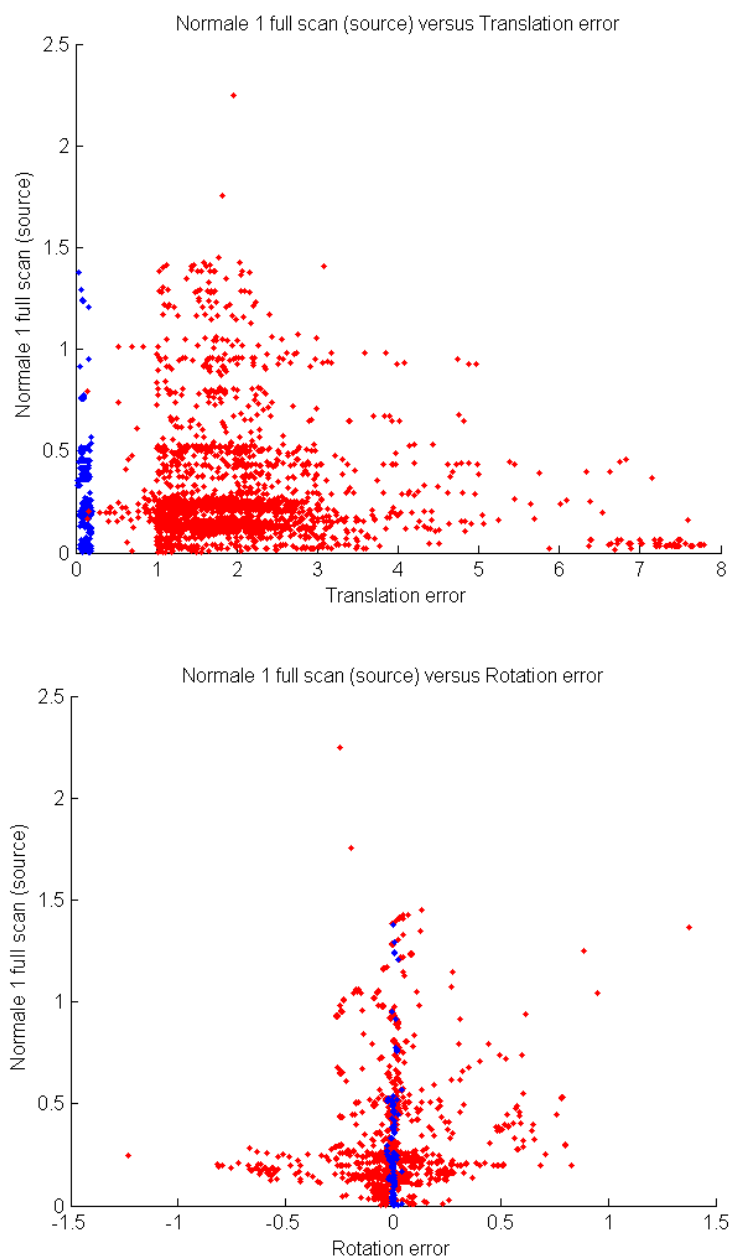


Figure A.7 Normale des points de tout le balayage source - méthode 1

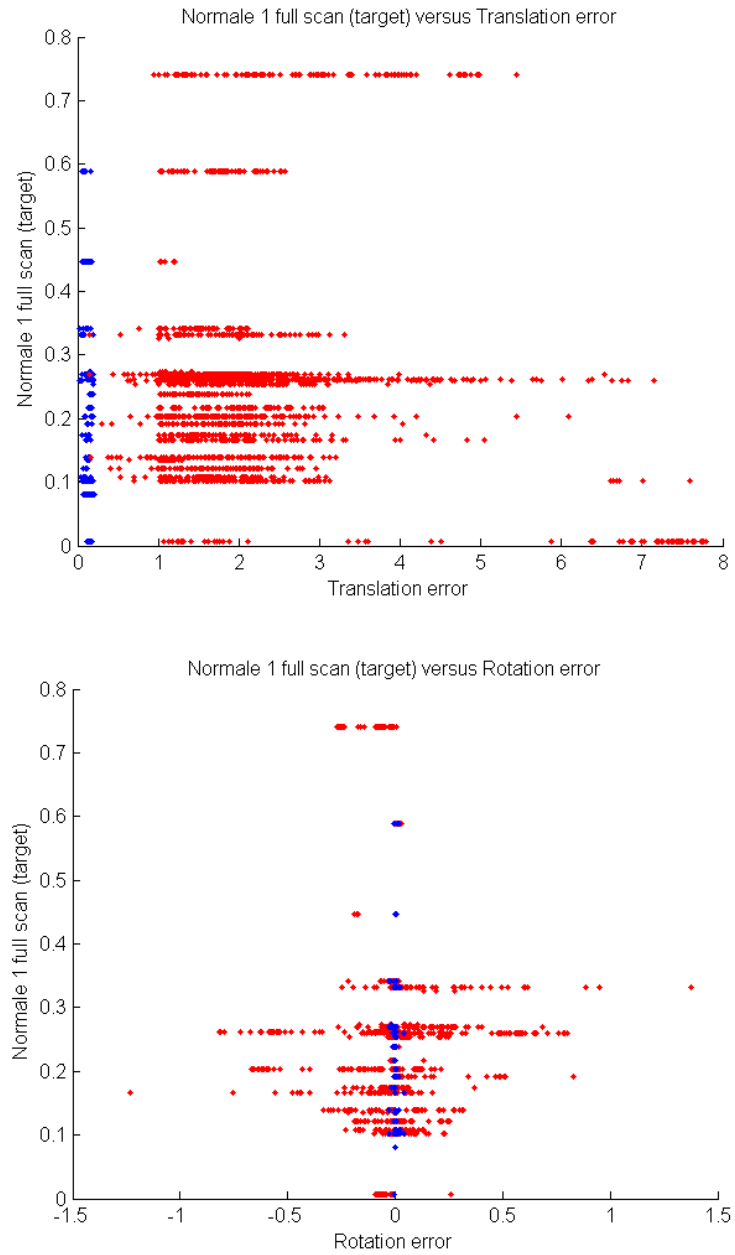


Figure A.8 Normale des points de tout le balayage cible - méthode 1

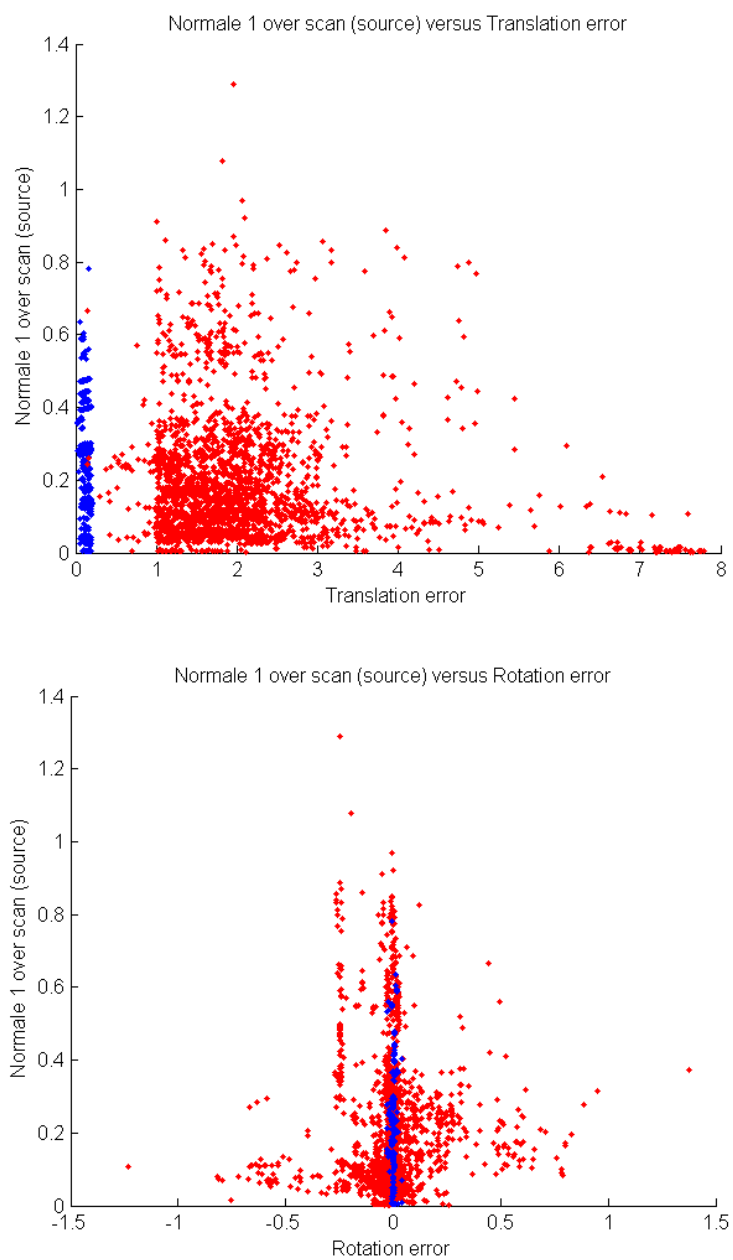


Figure A.9 Normale des points de la région de superposition du balayage source - méthode 1

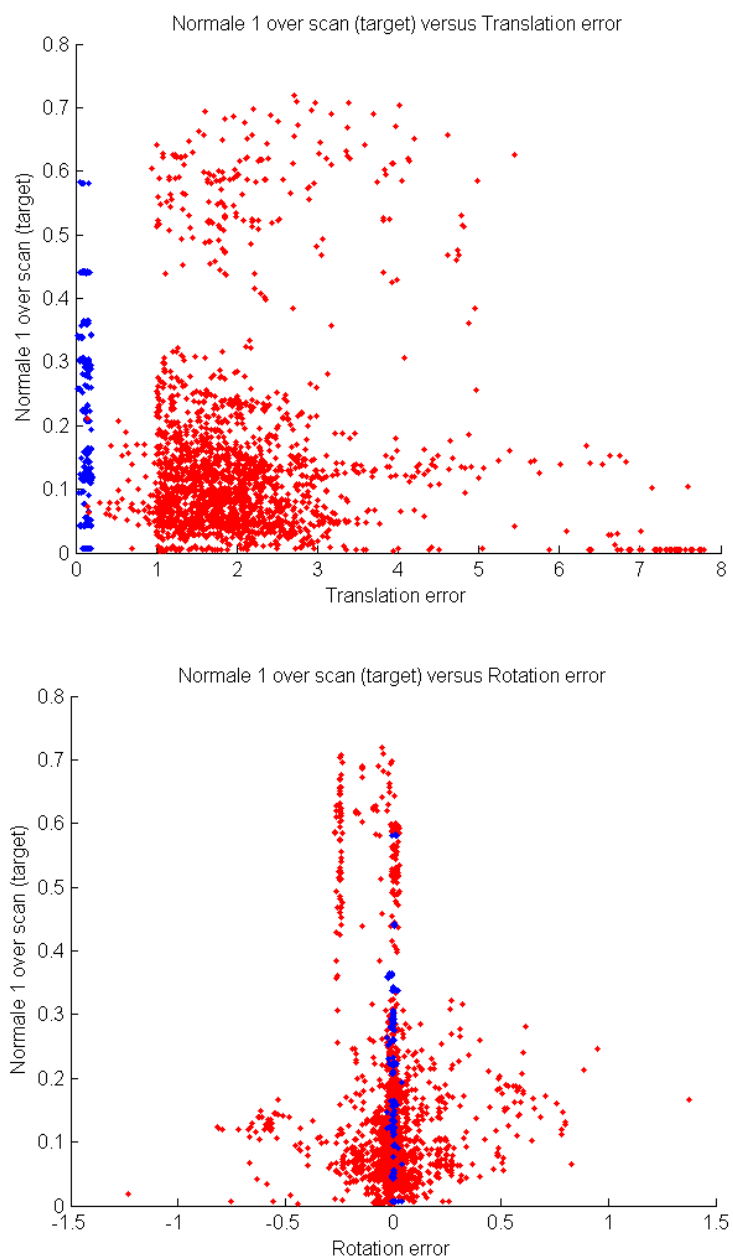


Figure A.10 Normale des points de la région de superposition du balayage cible - méthode 1



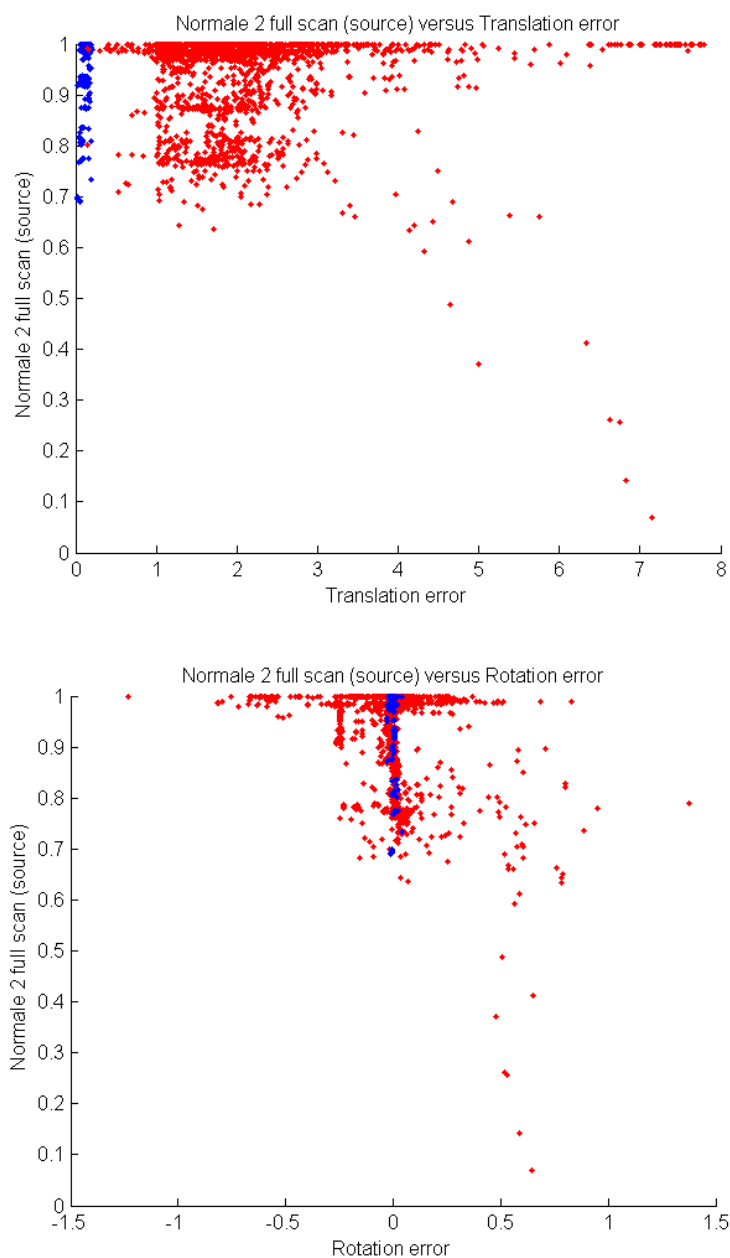


Figure A.11 Normale des points de tout le balayage source - méthode 2

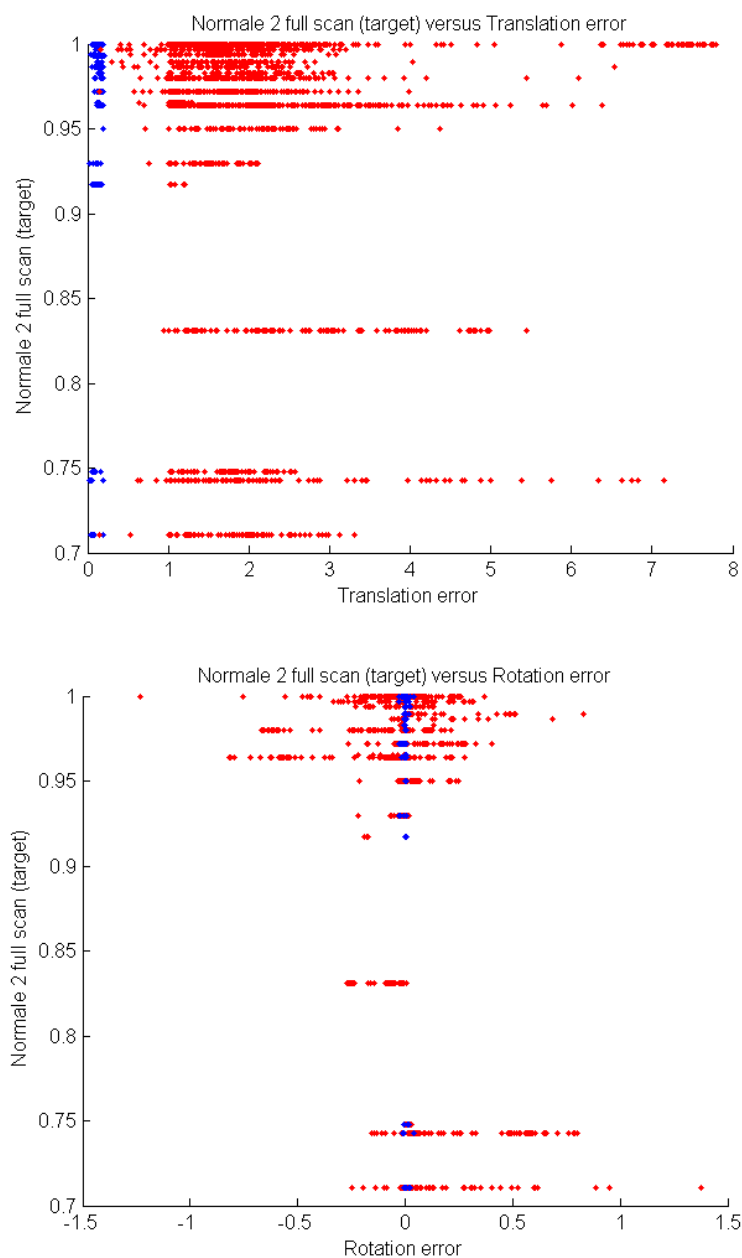


Figure A.12 Normale des points de tout le balayage cible - méthode 2

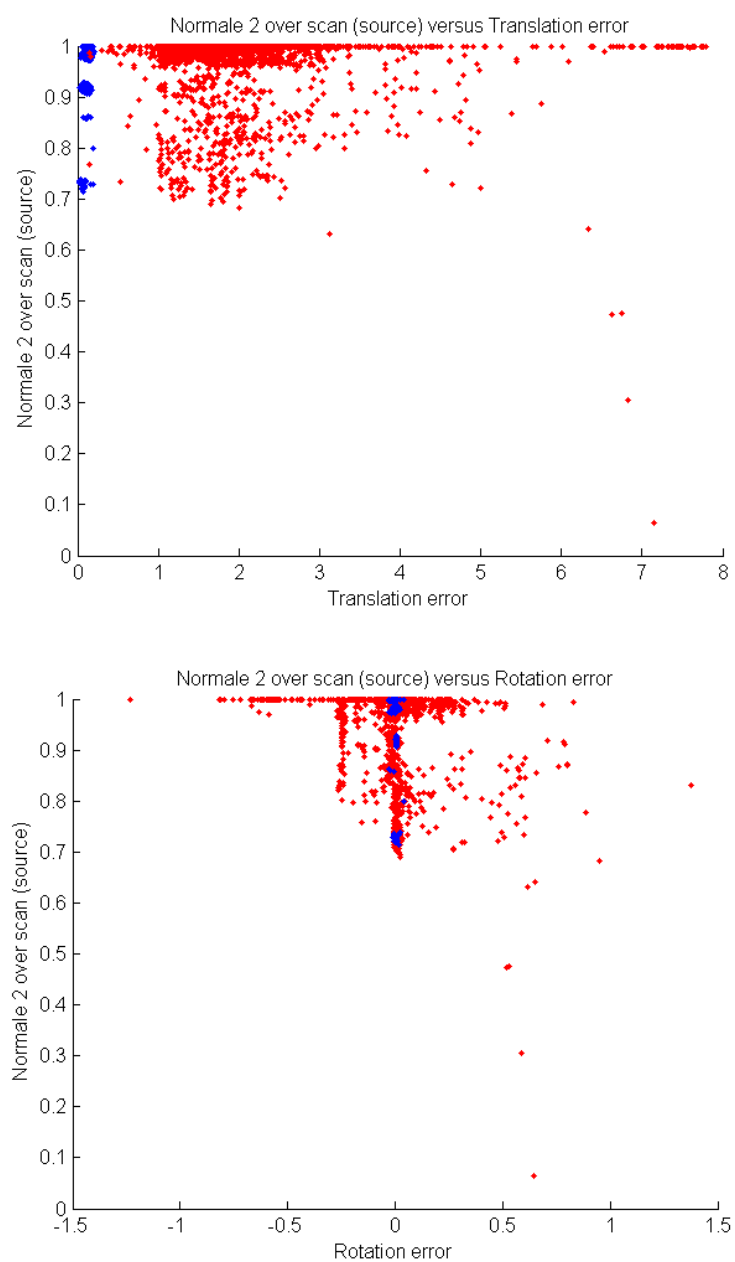


Figure A.13 Normale des points de la région de superposition du balayage source - méthode 2

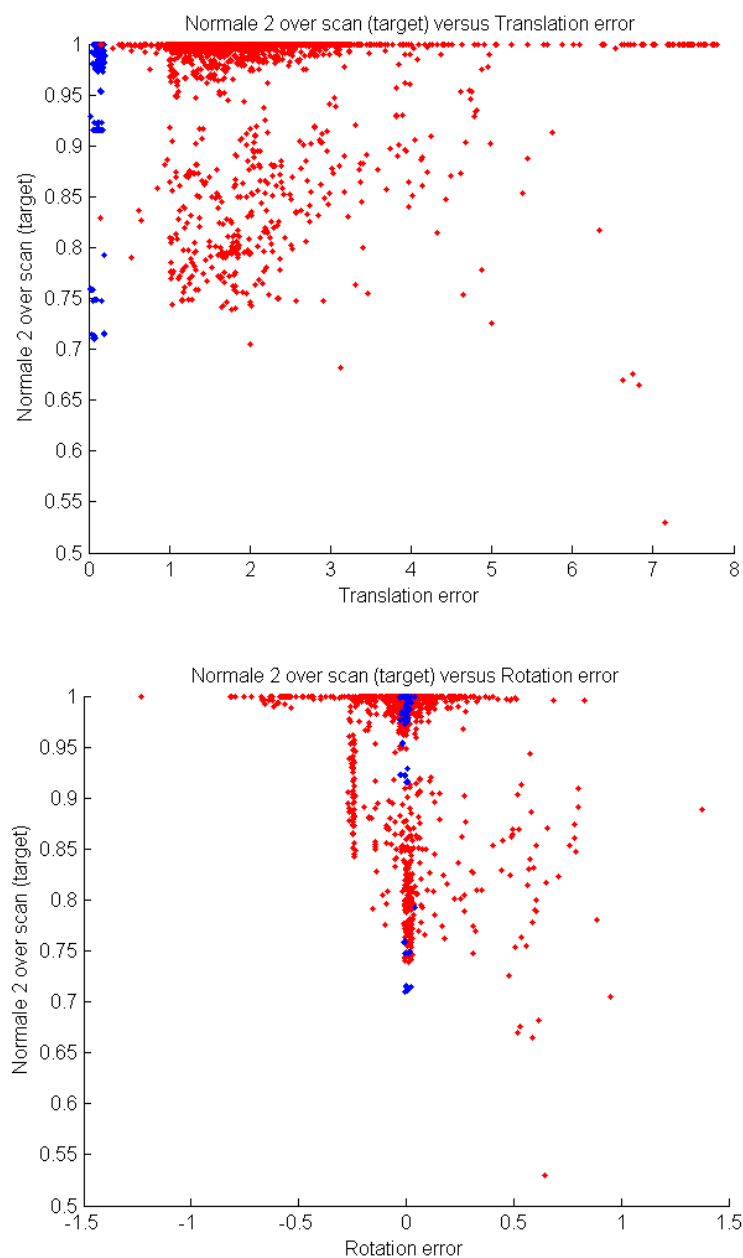


Figure A.14 Normale des points de la région de superposition du balayage cible - méthode 2

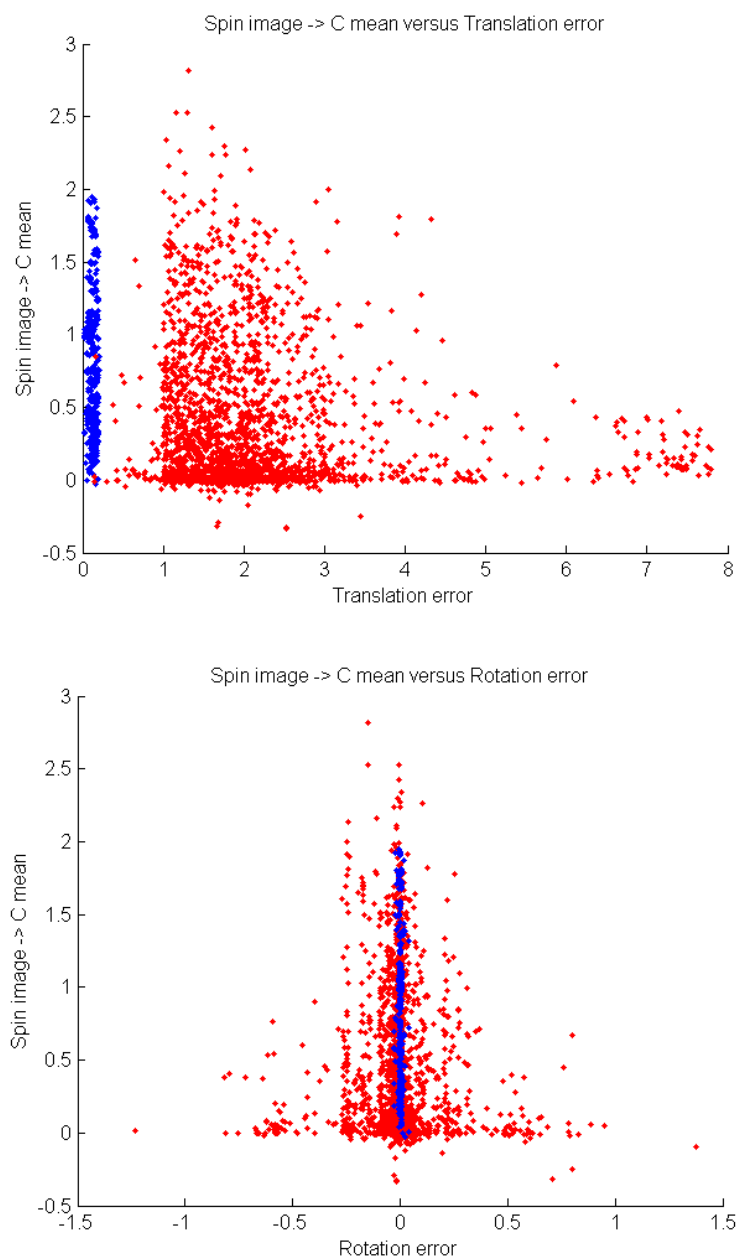


Figure A.15 Coefficient de corrélation des « spin images »

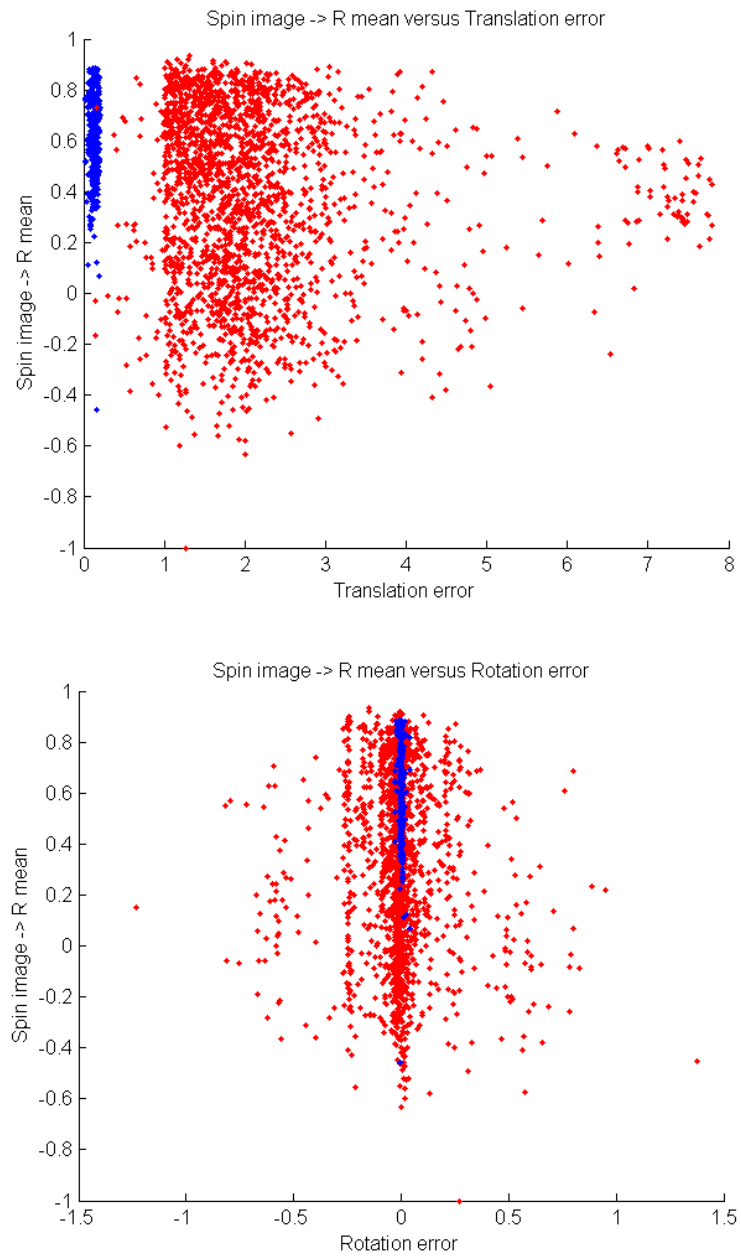


Figure A.16 Coefficient de confiance des « spin images »

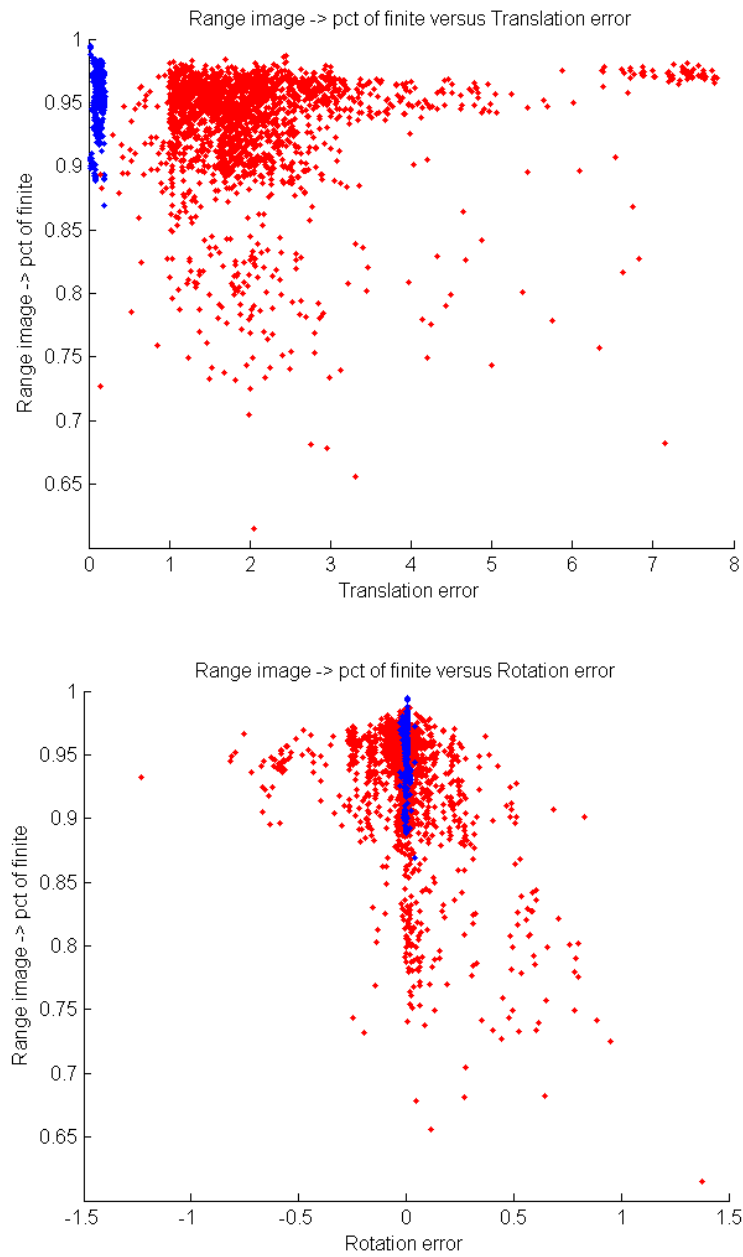


Figure A.17 Pourcentage de points finis des images de distance

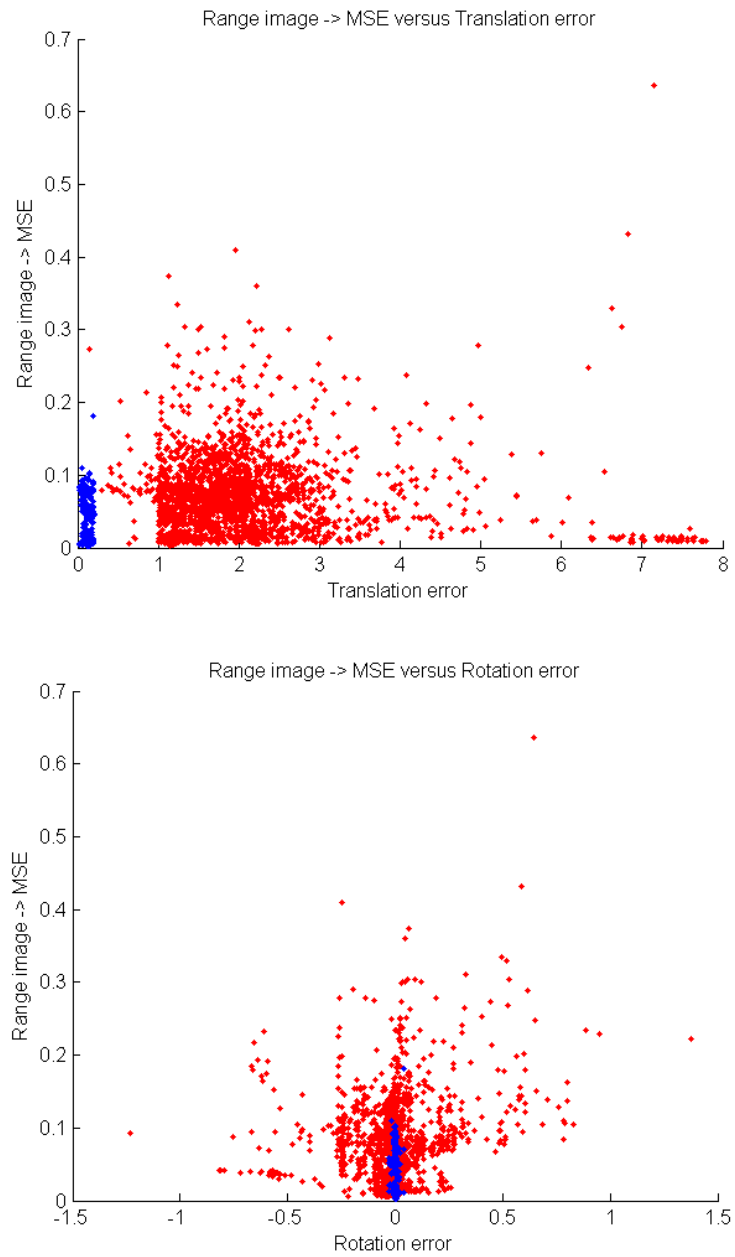


Figure A.18 Erreur quadratique moyenne des images de distance



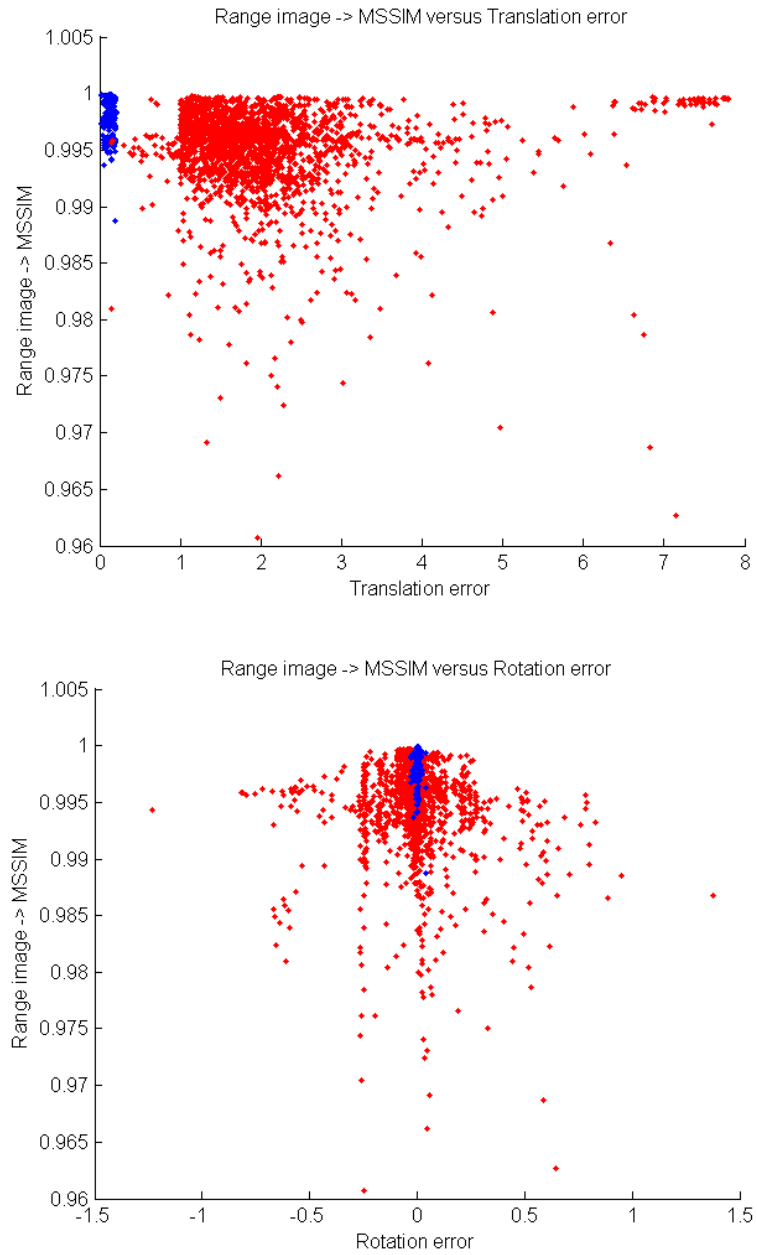


Figure A.19 MSSIM des images de distance

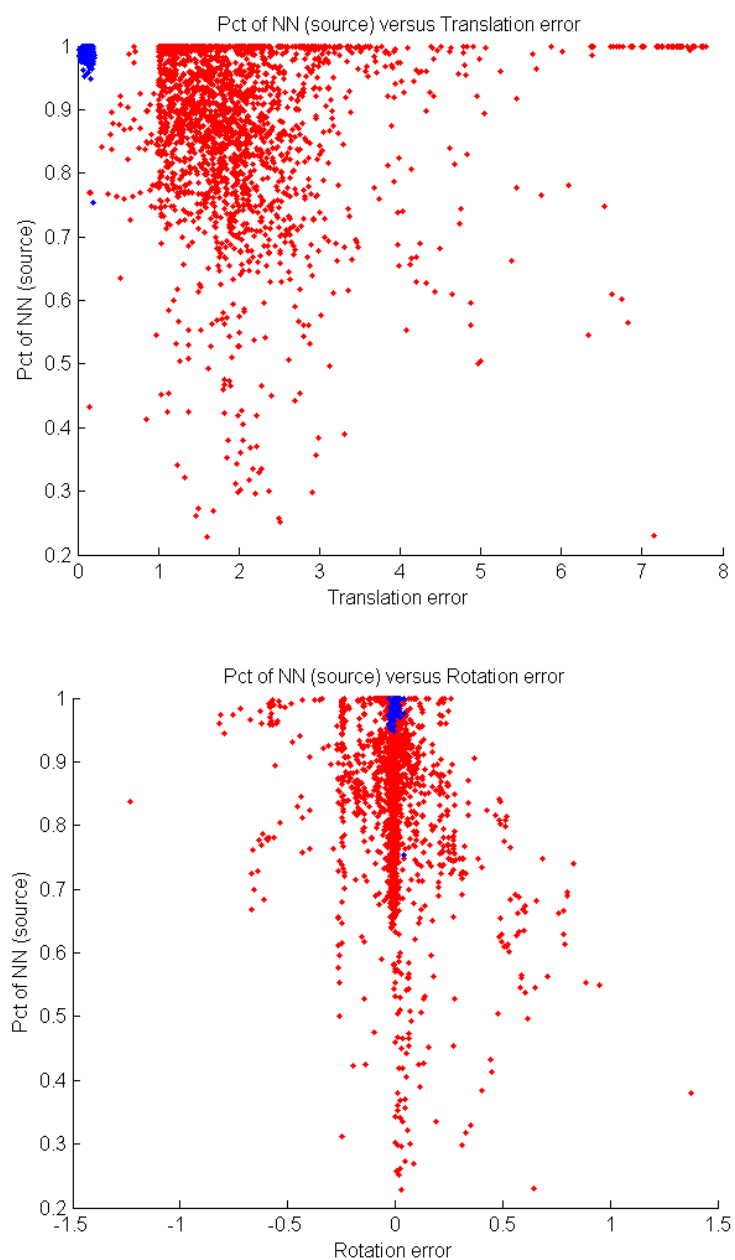


Figure A.20 Pourcentage du nombre de plus proches voisins du balayage source

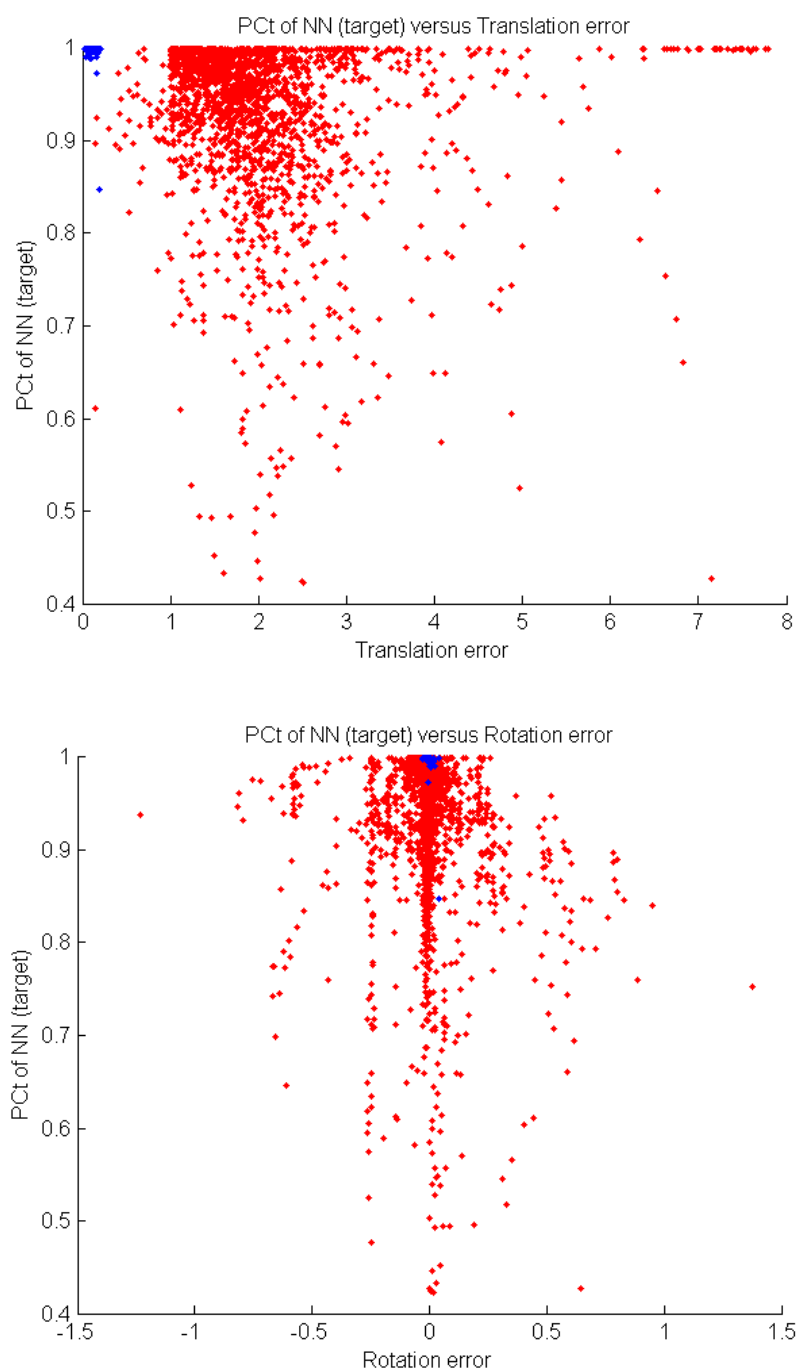


Figure A.21 Pourcentage du nombre de plus proches voisins du balayage cible